Analysis of Linkage-Friendly Genetic Algorithms

DISSERTATION

Laurence D. Merkle
Captain, USAF

AFIT/DS/ENG/96-11

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/DS/ENG/96-11

Analysis of Linkage-Friendly Genetic Algorithms

DISSERTATION

Laurence D. Merkle
Captain, USAF

AFIT/DS/ENG/96-11

Approved for public release; distribution unlimited

AFIT/DS/ENG/96-11

Analysis of Linkage-Friendly Genetic Algorithms

DISSERTATION

Presented to the Faculty of the Graduate School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Laurence D. Merkle, B.S., M.S.C.E.
Captain, USAF

December 1996

Dissertation

Analysis of Linkage-Friendly Genetic Algorithms

Capt Laurence D. Merkle
22 October 1996

Approved by

Committee Chairman: Gary B. Lamont
Professor Gary B. Lamont (PhD), AFIT/ENG

Committee Member: Mark E. Oxley
Professor Mark E. Oxley (PhD), AFIT/ENC

Committee Member: R Pachter
Adjunct Professor Ruth Pachter (PhD), WL/MLPJ

Committee Member:
Assistant Professor Eugene Santos, Jr. (PhD), AFIT/ENG

Dean's Representative: J. W. C
Associate Professor James W. Chrissis (PhD, PE), AFIT/ENS

Robert A. Calico, Jr.
Dean, Graduate School of Engineering

Acknowledgements

Many more people deserve my gratitude than I can possibly acknowledge here. The dissertation benefited greatly from the constructive criticism offered by my committee members, including Dr Eugene Santos and Dr James Chrissis. Responsibility for all remaining defects is of course mine alone. The courses I took from Dr Mark Oxley brought me from real analysis through optimization in infinite dimensional vector spaces, and gave me an appreciation of mathematical rigor. As a committee member, he cheerfully made time to indulge my often ill-posed questions. Dr Ruth Pachter introduced me to the protein folding problem, which has played a much larger role in my graduate education than is evidenced in this document. As my sponsor, she made certain that I had ample resources for my task. Most importantly, she gave me confidence in my own judgment when I needed it most.

Dr Gary Lamont introduced me to genetic algorithms, high performance computing, and many other areas of computer science. But in some sense, the subject matter is irrelevant. He coached me through the beginning of my journey through the "nonlinear process" of research. He always gave me plenty of room to stray from the path, so that I could learn to find it again for myself, but somehow kept me from becoming completely lost. He even made sure that I had some fun along the way. No student could ask for a better advisor.

My committee members are certainly not the only people with whom I have held enlightening discussions. George Gates and Charles Kaiser have done their best to keep me in touch with the "real" world as I "played" with my mathematical models. It has been a pleasure to learn from both of them. Hillol Kargupta graciously answered countless questions about the fast messy genetic algorithm, even when he had more important things to think about. Dave Goldberg turned out to be a really nice guy who was happy to hotly debate his latest theory one day, and help me find a great pair of running shoes the next. I appreciate the warm welcome he has always extended to me.

My parents, Doug and Carol, encouraged me to think freely at an early age, demonstrated the importance of hard work, and gave me a loving environment in which to grow. It would be hard to overstate the degree to which my father's example as an officer and a PhD has influenced me.

A close friend of mine whose husband recently completed his PhD told me that the acknowledgments are the most important part of the document. That may be true. but in no way can a few brief words express my appreciation for the support given to me by my wife Margaret. She has kept me fed and clothed. She has proofread numerous drafts. She has kept me company as I LaTeX'ed into the wee hours. She has endured my continual reinterpretation of current events in terms of computational theory and mathematical statistics. which may have been preferable to my more than occasional despondency. She has done her best to keep me focused, although I have not made that easy. She has forgiven a seemingly endless number of forgotten promises. Now she has willingly left behind the world she has known all of her life to live with me in unfamiliar surroundings. I hope that one day I can find better words. For now, I say "thank you."

Laurence D. Merkle

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

AFIT/DS/ENG/96-11

Abstract

Evolutionary algorithms (EAs) are stochastic population-based algorithms inspired by the natural processes of selection, mutation, and recombination. EAs are often employed as optimum seeking techniques. A formal framework for EAs is proposed, in which evolutionary operators are viewed as mappings from parameter spaces to spaces of random functions. Formal definitions within this framework capture the distinguishing characteristics of the classes of recombination, mutation. and selection operators. EAs which use strictly invariant selection operators and order invariant representation schemes comprise the class of linkage-friendly genetic algorithms (lfGAs).

Fast messy genetic algorithms (fmGAs) are lfGAs which use binary tournament selection (BTS) with thresholding. periodic filtering of a fixed number of randomly selected genes from each individual, and generalized single-point crossover. Probabilistic variants of thresholding and filtering are proposed. EAs using the probabilistic operators are generalized fmGAs (gfmGAs).

A dynamical systems model of lfGAs is developed which permits prediction of expected effectiveness. BTS with probabilistic thresholding is modeled at various levels of abstraction as a Markov chain. Transitions at the most detailed level involve decisions between classes of individuals. The probability of correct decision making is related to appropriate maximal order statistics. the distributions of which are obtained. Existing filtering models are extended to include probabilistic individual lengths.

Sensitivity of lfGA effectiveness to exogenous parameters limits practical applications. The lfGA parameter selection problem is formally posed as a constrained optimization problem in which the cost functional is related to expected effectiveness. Kuhn-Tucker conditions for the optimality of gfmGA parameters are derived. Parameter selection techniques are proposed for fmGAs and gfmGAs.

Analysis of Linkage-Friendly Genetic Algorithms

I. Introduction

1.1 Optimization

Many problems in science, engineering, and operations research may be viewed as *optimization problems*. Informally, an optimization problem involves a number of alternatives, and the solution to the problem is the set of alternatives which maximize or minimize some criterion. Examples include determination of the minimum energy state of a large biomolecule, design of a jet engine with maximum thrust-to-weight ratio, and minimization of the total distance traveled in visiting a set of destinations. Frequently, the alternatives are also subject to one or more constraints. In the present examples, certain atoms of the biomolecule may form a ring system, there may be a maximum allowable cost for the engine, and repeated visits to a particular destination may be prohibited. This last example is the famous *Traveling Salesperson Problem (TSP)* [26]. Optimization problems are defined formally in Chapter II.

Often, the set of alternatives is isomorphic to a region[1] (or union of several regions) of $\mathbb{R}^n$. For example, in identifying the minimum energy state of a large biomolecule, the possible arrangements of the constituent atoms are determined by some subset of the molecule's dihedral angles (making the common simplifying assumption that all bond lengths and bond angles, and possibly some of the dihedral angles, are fixed at their "equilibrium" values [73]). Each angle takes on values in the interval $(-\pi, \pi] \subset \mathbb{R}$, so the set of possible states is isomorphic to $(-\pi, \pi]^n \subset \mathbb{R}^n$ where n is the number of variable dihedral angles.

In contrast, the set of alternatives for a *combinatoric* optimization problem is discrete (i.e. finite or countably infinite). The previous example may be further simplified by assuming that each variable dihedral angle assumes values in $\{-\frac{\pi}{3}, \frac{\pi}{3}, \pi\}$, so that the set of alternatives is isomorphic to $\{-\frac{\pi}{3}, \frac{\pi}{3}, \pi\}^n$. The resulting

[1] Apostol [3] defines "region" as follows: "A set in $\mathbf{R}^n$ is called a region if it is the union of an open connected set with some, none, or all its boundary points."

optimization problem is combinatoric. Likewise, the set of alternatives in the TSP is the set of permutations over the set of destinations. This set is of course finite, although possibly quite large.

There is only one general method guaranteed to obtain the optimum alternative of an arbitrary optimization problem, that being exhaustive search [63] (pure random search being a special case thereof). For many optimization problems of practical interest, the number of alternatives prohibits their enumeration. Thus the practitioner is often forced to settle for solutions which are "good enough," even though they are not optimal. When the problem is posed as such, it may be referred to as a *semi-optimization* problem [62], and the solution techniques employed are called *optimum-seeking techniques.* Even finding an "acceptable" solution may require exploring a significant part of a large search space.

The various techniques used to solve optimization problems are part of what come to be a unified theory of optimization. Some techniques have been known and used for centuries, while the advent of the high speed digital computer has enabled the application of other techniques which were previously impractical due to the large number of calculations required. It also brought about the development of entirely new techniques designed explicitly to exploit the strengths of the digital computer. Still, problem sizes are limited by processor speed and memory size, and the relative efficiency and effectiveness of existing algorithms are not necessarily preserved by hardware advances.

This last observation is especially important as multiprocessor architectures (parallel, distributed, or otherwise) become more prevalent. Realization of the effectiveness and efficiency improvements made possible by such architectures depends on the design and use of appropriate multiprocessor algorithms. Just as the arrival of single processor computers spawned the development of new techniques, multiprocessor architectures invite the investigation of a new set of optimization tools.

1.2 Evolutionary Algorithms

A promising set of candidates for such investigation are a class of algorithms inspired by the principles of evolution, known appropriately as *evolutionary algorithms.* These techniques operate by applying biologically-inspired operators, such as recombination, mutation, and selection, to a population of individuals, each of which represents a candidate solution (alternative). Their use as optimum seeking techniques derives from the resulting analogy to the principle of "survival of the fittest."

Because of their population based approach, evolutionary algorithms are well suited for implementation on multiprocessor systems. One example which receives significant attention in the literature in this regard is the *simple genetic algorithm.* The *fast messy genetic algorithm* is another. It is representative of a class of evolutionary algorithms which are called *linkage-friendly genetic algorithms* in this research. Each of these algorithms is defined precisely in Chapter II.

Linkage-friendly genetic algorithms are potentially more effective and efficient than the simple genetic algorithm for a large class of optimization problems, but their properties are not well understood. In particular, the fast messy genetic algorithm is demonstrated in limited applications to pedagogical problems [35] to be an effective and efficient optimum seeking technique. However, its practical use is limited by its dependence on a large number of exogenous parameters. Specifically, its effectiveness depends strongly on its many "filtering" and "thresholding" parameters. Currently available parameter selection methodologies [35, 46, 47] are *ad hoc* in nature and do not reliably result in satisfactory effectiveness.

1.3 Problem Statement and Approach

The primary objectives of this research are to

- mathematically model those properties of specific linkage-friendly genetic algorithms which are related to expected effectiveness; and
- develop exogenous parameter selection techniques for those linkage-friendly genetic algorithms, focusing on maximizing their expected effectiveness.

Linkage-friendly genetic algorithms are modeled as dynamical systems, and expected effectiveness is defined as a deterministic function of the system state. The state transitions are determined by the specific operators used by the algorithm, and the parameters of those operators. The operators modeled are generalizations of the fast messy genetic algorithm's "building block filtering" and "binary tournament selection with thresholding" operators.

The dynamical systems model predicts the expected effectiveness resulting from a particular choice of filtering and thresholding parameters. Consequently, the parameter selection problem may be posed as an optimization problem. The set of alternatives is the permissible set of filtering and thresholding parameters, and the criterion to be maximized is expected effectiveness. Taking this perspective, this research develops exogenous parameter selection techniques based on standard optimum seeking techniques. A parameter selection technique is considered acceptable if it satisfies the following criteria:

1. the technique guarantees expected effectiveness no worse than that resulting from the best set of parameters obtained using existing techniques,
2. the technique requires no *a priori* knowledge of the optimal solution,
3. the technique requires no design parameters beyond those of the linkage-friendly genetic algorithm; and
4. the computational effort required by the technique scales well with the effort required by the linkage-friendly genetic algorithm.

1.4 Organization of the Dissertation

The background necessary to fully define the problem outlined in Section 1.3 is provided in Chapter II. beginning with a brief introduction to optimization theory and development of a general framework for evolutionary algorithms. The remainder of the chapter introduces simple genetic algorithms, then focuses on linkage-friendly genetic algorithms. Chapter III presents the previously mentioned generalizations of the fast messy genetic algorithm operators. The mathematical model of building block filtering is developed in Chapter IV, and the model of tournament selection is developed in Chapter V. The parameter selection problem is formally posed as an optimization problem and parameter selection techniques are proposed in Chapter VI. Finally. Chapter VII presents conclusions and recommendations for future research.

II. Selected Topics in Evolutionary Algorithms

This chapter provides a background in evolutionary algorithms (EAs), focusing on genetic algorithms (GAs). Selected concepts of optimization theory are introduced in Section 2.1. The relationship of GAs to the more general class of EAs is discussed briefly in Section 2.2. A considerable portion of the chapter (Section 2.3) is devoted to development of a formal framework for evolutionary algorithms, including aspects which are novel contributions of this research. The section presents standard definitions of *decoding* and *fitness scaling functions*, as well as novel definitions of *evolutionary operators* in general and *recombination*, *mutation*, and *selection operators* in particular.

Within the formal framework, Section 2.4 defines the *simple genetic algorithm* (sGA) [29], which is a primary focus in EA research. As an optimum seeking technique, the sGA exhibits the significant drawback that its effectiveness is sensitive to both permutations in the decoding function and the choice of fitness scaling function. The dependence of sGA effectiveness on the fitness scaling function is examined theoretically in Section 2.5. The remainder of the chapter discusses linkage-friendly genetic algorithms (Section 2.6), for which the effectiveness is independent of both permutations in the decoding function and the choice of fitness scaling function.

2.1 Optimization Theory

This section introduces selected fundamental concepts and terminology of optimization theory. For a more thorough treatment, including discussion of techniques, see for example Pierre [63]. An optimization problem involves either maximization or minimization of a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ over a set $\Omega \subseteq \mathbb{R}^n$. The function f is known by various names in the optimization literature including *objective function*, *objective functional*, *cost function*, and *performance measure*. The set Ω is known as either the *feasible region*, the *feasible set*, or the *admissible set*.

A point $\hat{\mathbf{x}} \in \Omega$ is called a *point of strong local maximum* (or *maximizer*) if there exists an $\epsilon > 0$ such that if $\mathbf{x} \in \Omega - \{\hat{\mathbf{x}}\} \neq \{\}$ and $||\mathbf{x} - \hat{\mathbf{x}}|| < \epsilon$ then $f(\hat{\mathbf{x}}) > f(\mathbf{x})$. In the sequel, such a point $\hat{\mathbf{x}}$ is called simply a *point of local maximum.* The value $f(\hat{\mathbf{x}})$ is then a *local maximum* of f in Ω. Definitions of *points of (strong) local minimum* and *local minimum* are of course directly analogous. Any point $\hat{\mathbf{x}} \in \Omega$ which is either a point of local maximum or a point of local minimum is called a *relative extremum point.*

If a point $\hat{\mathbf{x}} \in \Omega$ satisfies $f(\hat{\mathbf{x}}) > f(\mathbf{x})$ for *all* $\mathbf{x} \in \Omega - \{\hat{\mathbf{x}}\} \neq \{\}$, then $f(\hat{\mathbf{x}})$ is the *absolute maximum* (or *global maximum*). Similarly, if a point $\hat{\mathbf{x}} \in \Omega$ satisfies $f(\hat{\mathbf{x}}) < f(\mathbf{x})$ for *all* $\mathbf{x} \in \Omega - \{\hat{\mathbf{x}}\} \neq \{\}$, then $f(\hat{\mathbf{x}})$ is the *absolute minimum* (or *global minimum*).

2.2 Relationship of Genetic Algorithms to Evolutionary Algorithms

Genetic algorithms (GAs) are a form of computation inspired by theories of evolution. This places them in the class of algorithms called Evolutionary Algorithms (EAs). Other members of this class include Evolution Strategies (ESs) [66, 69] and Evolutionary Programming (EP) [24] (see Figure 1). Thomas Bäck [6] provides an excellent review of all three. including a historical perspective.

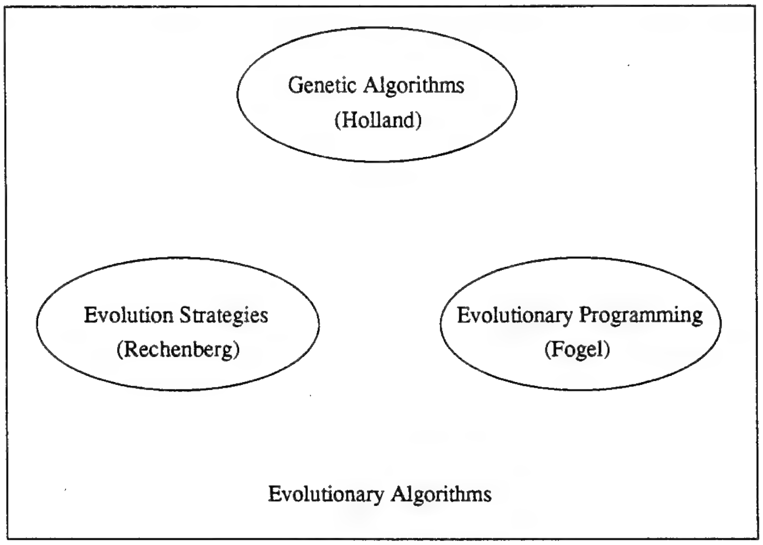


Figure 1. Venn Diagram of the Class of Evolutionary Algorithms

Genetic algorithms were first proposed by Holland in connection with his theories of complex adaptive systems [42] (where they were called "reproductive plans"). De Jong later applied genetic algorithms to the functional optimization problem [17]. Since then, a significant portion of the genetic algorithm literature has been devoted to modifications aimed at improving the effectiveness and efficiency of genetic algorithms as optimum seeking techniques.

Historically, the three major areas developed independently between the 1960's and the 1980's. Interest in evolutionary algorithms, and genetic algorithms in particular, grew dramatically in the late 1980's and early 1990's, as demonstrated by the success of numerous international conferences (see Table 1) and the appearance of a number of textbooks (e.g. [6, 24, 29, 44, 58, 60, 70]). Also during this period, interaction between the communities increased, as many researchers began transferring analytical insight and experimental approaches amongst the three fields.

Table 1. Major Evolutionary Algorithm Conference Proceedings

Conference	Primary Focus	Proceedings
International Conference on Genetic Algorithms	GAs	[37, 38, 67, 8, 25, 19]
Foundations of Genetic Algorithms	GAs	[65, 75, 76, 9]
Parallel Problem Solving from Nature	GAs and ESs	[68, 51, 13, 18]
Annual Conference on Evolutionary Programming	EPs	[20, 21, 71, 52, 23]
IEEE International Conference on Evolutionary Computing	EAs	[59, 22, 72]

2.3 A Framework for Evolutionary Algorithms

As a first step towards unifying the theory of the three major evolutionary algorithm paradigms, Bäck and Schwefel propose a general "algorithmic description" for EAs, which they specialize for each of the three paradigms [7]. The various mappings appearing in their description are defined so broadly that their essential characteristics are overlooked. This section develops formal definitions of the mappings which capture these characteristics, then presents an extension of Bäck and Schwefel's algorithmic description.

2.3.1 Representation. Associated with each evolutionary algorithm is a non-empty set I, called the *individual space* of the algorithm. Each *individual* $\mathbf{a} \in I$ represents a candidate solution to the optimization problem at hand. The representation scheme is formally defined by the *decoding function*.

Definition 2.3.1 (Decoding function): *Let I be a non-empty set (the* individual space*). and $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ (the* objective function*). If $D : I \longrightarrow \mathbb{R}^n$ is total. i.e. the domain of D is all of I. then D is called a* decoding function. □

The mapping D is not necessarily surjective (in fact. it cannot be if I is countable). The range of D determines the subset of $\mathbb{R}^n$ actually available for exploration by the evolutionary algorithm.

The *fitness* of an individual is an indication of the quality of the candidate solution represented by the individual. The mapping which yields this indication is the *fitness function.* It is the fitness function which the evolutionary algorithm actually attempts to optimize.

Definition 2.3.2 (Fitness function): *Let I be a non-empty set (the individual space), $D : I \longrightarrow \mathbb{R}^n$ (the decoding function), $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ (the objective function). and $T_s : \mathbb{R} \longrightarrow \mathbb{R}$ (the fitness scaling function[1]). Then $\Phi \overset{\triangle}{=} T_s \circ f \circ D$ is called a* fitness function. □

In this definition it is understood that the objective function f is determined by the application. while the specification of the decoding function D and the fitness scaling function T_s are design issues. An important design criteria for the scaling function is that it preserve the partial ordering induced on the individual space by the decoding and objective functions.

Definition 2.3.3 (Order-preserving fitness scaling function): *If for every non-empty set I (the individual space). every $(\mathbf{a}, \mathbf{b}) \in I^2$. every $D : I \longrightarrow \mathbb{R}^n$ (the decoding function), and every $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ (the objective function). a mapping $T_s : \mathbb{R} \longrightarrow \mathbb{R}$ (the fitness scaling function) satisfies*

$$f(D(\mathbf{a})) < f(D(\mathbf{b})) \Longrightarrow T_s(f(D(\mathbf{a})) < T_s(f(D(\mathbf{b}))) \ ,$$

then T_s is called an order-preserving fitness scaling function. □

[1] Use of the term "scaling" for the mapping T_s is consistent with the genetic algorithms literature. although is not descriptive of many of the operators used in practice.

The following lemma provides a condition which is necessary and sufficient for a scaling function to be order-preserving.

Lemma 2.3.4 *A fitness scaling function is order-preserving if and only if it is strictly increasing.*

Proof: "If": Let $T_s : \mathbb{R} \longrightarrow \mathbb{R}$ be strictly increasing, I a non-empty set, $(\mathbf{a}, \mathbf{b}) \in I^2$, $D : I \longrightarrow \mathbb{R}^n$ for some $n \in \mathbb{N}$, $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $g_a \triangleq f(D(\mathbf{a}))$, and $g_b \triangleq f(D(\mathbf{b}))$. Suppose $f(D(\mathbf{a})) < f(D(\mathbf{b}))$. Then $g_a < g_b$, and because T_s is strictly increasing, $T_s(f(D(\mathbf{a}))) = T_s(g_a) < T_s(g_b) = T_s(f(D(\mathbf{b})))$. Thus, $f(D(\mathbf{a})) < f(D(\mathbf{b})) \Longrightarrow T_s(f(D(\mathbf{a})) < T_s(f(D(\mathbf{b}))$. Since I, $\mathbf{a}$, $\mathbf{b}$, D, and f are arbitrary, T_s is an order-preserving fitness scaling function.

"Only if": Let $T_s : \mathbb{R} \longrightarrow \mathbb{R}$ be an order-preserving fitness scaling function, $x, y \in \mathbb{R}$, $I \triangleq \{x, y\}$, $\mathbf{a} \triangleq x$, $\mathbf{b} \triangleq y$, and $D = f : \mathbb{R} \longrightarrow \mathbb{R}$ the identity mapping. Suppose $x < y$. Then $f(D(\mathbf{a})) = x < y = f(D(\mathbf{b}))$, and because T_s is an order-preserving fitness scaling function $T_s(x) = T_s(f(D(\mathbf{a}))) < T_s(f(D(\mathbf{b}))) = T_s(y)$. Thus, $x < y \Longrightarrow T_s(x) < T_s(y)$. Since x and y are arbitrary, T_s is strictly increasing. ■

Execution of an evolutionary algorithm typically begins by randomly sampling individuals from I. The sampling is typically performed with replacement, and the resulting collection of individuals is called the *initial population*, denoted $P(0)$.[2] More generally, a *population* is a collection $P = \{\mathbf{a}_1, \ldots, \mathbf{a}_\mu\}$ of individuals $\mathbf{a}_i \in I$, and the number of individuals μ in the population is the *population size.*

Following initialization, execution proceeds iteratively. Each iteration consists of application of one or more *evolutionary operators.* The combined effect of the evolutionary operators applied in a particular *generation* $t \in \mathbb{N}$ is to transform the current population $P(t)$ into a new population $P(t+1)$.

2.3.2 Evolutionary Operators. Most authors, including Bäck and Schwefel, describe evolutionary operators as directly mapping populations into populations, with the mapping being "controlled" by the

[2] In this research, populations are treated interchangeably as n-tuples of individuals or multisets of individuals, as convenient. The term "multiset" describes the primitive mathematical concept of a collection of elements for which the multiplicities of the elements, but not their order, is important [1]. For example, the multiset $\{a, a, b\}$ is equal to the multiset $\{a, b, a\}$, and neither is equal to the multiset $\{a, a, b, b\}$. Multisets are also called *bags.*

One further word regarding terminology. What is referred to as a "population" in the evolutionary algorithms literature is referred to as a "sample" in the mathematical statistics literature. Similarly, the "individual space" of evolutionary algorithms corresponds to the "population" of mathematical statistics. The latter is also called the "grand population" [12].

parameters of the operator. This research proposes a more formal view of evolutionary operators as mappings from parameter spaces to random population transformations (i.e., random functions[3] with values in the set of population transformations). This view precisely identifies the relationships among the operator parameters and the various mappings. In the following definitions and the sequel, the set of mappings from a set $\mathcal{S}_1$ to a set $\mathcal{S}_2$ is denoted $\mathcal{T}(\mathcal{S}_1, \mathcal{S}_2)$.

The first definition is that of a population transformation, which is any mapping from populations to populations, whether or not the populations are of the same size (see Figure 2).

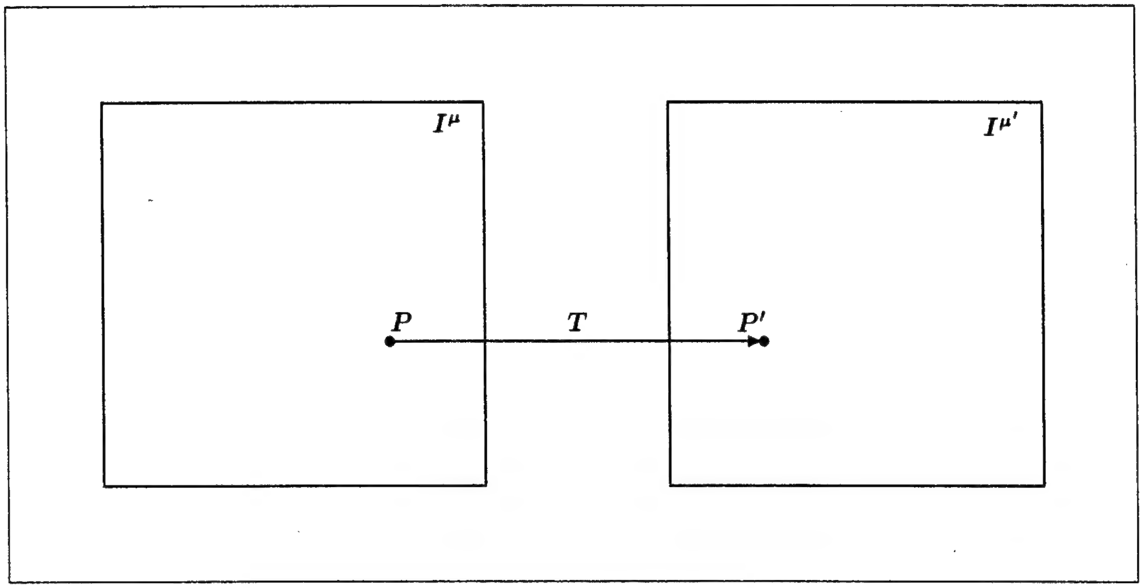


Figure 2. The population transformation T deterministically maps the parent population P (of size μ) to the offspring population P' (of size μ').

Definition 2.3.5 (Population transformation): *Let I be a non-empty set (the individual space), and $\mu, \mu' \in \mathbb{Z}^+$ (the* parent *and* offspring population sizes, *respectively). A mapping $T : I^\mu \longrightarrow I^{\mu'}$ is called a* population transformation. *If $T(P) = P'$ then P is called a* parent population *and P' is called an* offspring population. *If $\mu = \mu'$, then they are called simply the* population size. □

[3] Let Ω be a sample space, and let $\mathcal{V}$ be a set of functions. Then $X : \Omega \longrightarrow \mathcal{V}$ is a random function [10]. The most frequently encountered random functions are *stochastic processes*, for which the domain of each $f \in \mathcal{V}$ is $\mathbb{R}$.

The population transformation resulting from the application of an evolutionary operator, to include the offspring population size, often depends on the outcome of a random experiment. This dependence motivates the concept of a random population transformation (Figure 3).

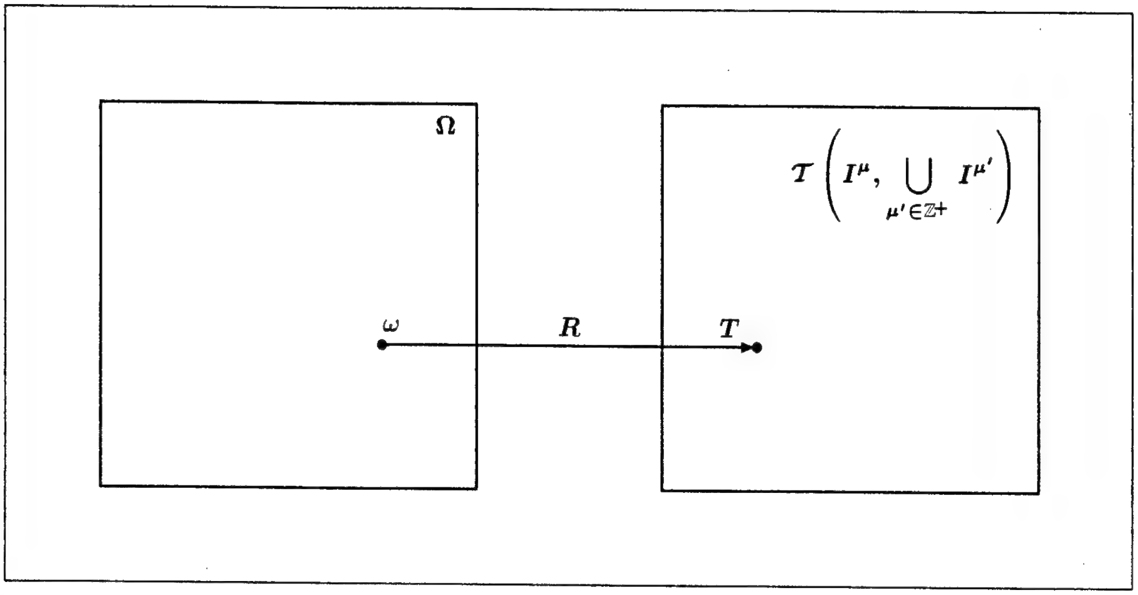

Ω
$\mathcal{T}\left(I^{\mu}, \bigcup_{\mu' \in \mathbb{Z}^+} I^{\mu'}\right)$
ω
R
T

Figure 3. The random population transformation R maps the random event ω (with sample space Ω) to the population transformation T, which maps parent populations of size μ (which is independent of ω) to offspring populations of some fixed size $\mu' \in \mathbb{Z}^+$ (which may depend on ω).

Definition 2.3.6 (Random population transformation): *Let I be a non-empty set (the individual space), $\mu \in \mathbb{Z}^+$ (the parent population size), and Ω a set (the* sample space*). A random function*

$$R : \Omega \longrightarrow \mathcal{T}\left(I^{\mu}, \bigcup_{\mu' \in \mathbb{Z}^+} I^{\mu'}\right)$$

is called a random population transformation. □

The distribution of population transformations resulting from the application of an evolutionary operator may depend on one or more parameters of the operator. That is, each evolutionary operator maps its parameters to a random population transformation (Figure 4).

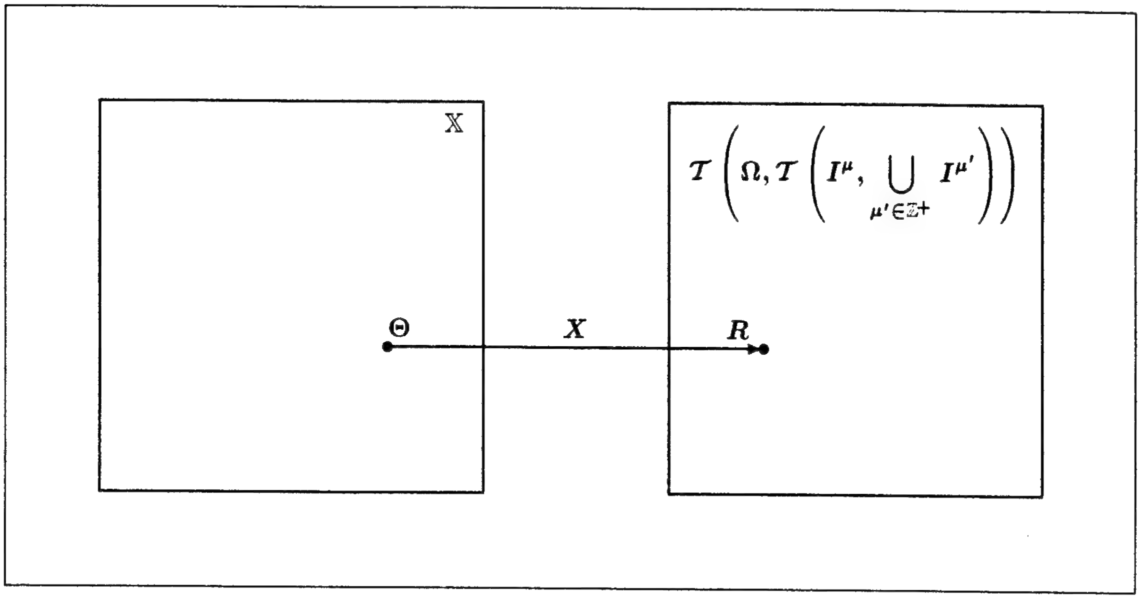

$\mathbb{X}$
$\mathcal{T}\left(\Omega, \mathcal{T}\left(I^{\mu}, \bigcup_{\mu' \in \mathbb{Z}^+} I^{\mu'}\right)\right)$
Θ
X
R

Figure 4. The evolutionary operator X maps the exogenous parameter(s) Θ to the random population transformation R. The underlying sample space of R is Ω. Each of the possible population transformations acts on populations of size μ. The offspring population size $\mu' \in \mathbb{Z}^+$ may depend on Θ as well as the random event $\omega \in \Omega$.

Definition 2.3.7 (Evolutionary operator): *Let I be a non-empty set (the individual space), $\mu \in \mathbb{Z}^+$ (the parent population size), $\mathbb{X}$ a set (the* parameter space*), and Ω a set (the* sample space*). A mapping*

$$X : \mathbb{X} \longrightarrow \mathcal{T}\left(\Omega, \mathcal{T}\left(I^{\mu}, \bigcup_{\mu' \in \mathbb{Z}^+} I^{\mu'}\right)\right) \tag{1}$$

is called an evolutionary operator. *The set of evolutionary operators in the form of Equation 1 is denoted $\mathcal{EVOP}(I, \mu, \mathbb{X}, \Omega)$.* □

The random population transformation $X(\Theta)$ is denoted X_Θ. The population transformation $X_\Theta(\omega)$ is also denoted X_Θ to maintain consistency with the notation of Bäck and Schwefel, except where confusion may arise. In particular, the offspring population $[X_\Theta(\omega)](P)$ is denoted $X_\Theta(P)$. Finally, if X has no parameters, i.e. $X \in \mathcal{EVOP}(I, \mu, \{\}, \Omega)$, then the offspring population is denoted $X(P)$.

The specific evolutionary operators used are typically biologically inspired. The guiding principle in their design is typically loose analogy to Darwin's principle of "survival of the fittest." The most commonly used evolutionary operators are *recombination*, *mutation*, and *selection*.

Recombination operators are the most general of the three. The distinguishing characteristic of recombination operators is that at least some of the individuals in the offspring population may depend on more than one individual in the parent population. The following definition reflects this characteristic. Because of this, it is more restrictive than the overly general definition adopted by Bäck and Schwefel, which admits any population transformation $r : I^{\mu} \longrightarrow I^{\mu'}$ where $\mu, \mu' \in \mathbb{Z}^+$.

Definition 2.3.8 (Recombination operator): *Let $r \in \mathcal{EVOP}(I, \mu, \mathbb{X}, \Omega)$. If there exist $P \in I^{\mu}$, $\Theta \in \mathbb{X}$, and $\omega \in \Omega$ such that at least one individual in the offspring population $r_{\Theta}(P)$ depends on more than one individual of P then r is called a* recombination operator. □

In contrast to recombination operators, the distinguishing feature of mutation operators is that each of the individuals in the offspring population depends on at most one individual in the parent population.

Definition 2.3.9 (Mutation operator): *Let $m \in \mathcal{EVOP}(I, \mu, \mathbb{X}, \Omega)$. If for every $P \in I^{\mu}$, every $\Theta \in \mathbb{X}$, and every $\omega \in \Omega$, each individual in the offspring population $m_{\Theta}(P)$ depends on at most one individual of P then m is called a* mutation operator. □

This definition of mutation is more general than Bäck and Schwefel's, which assumes that parent and offspring population sizes are equal.

The distinguishing characteristics of selection operators are that every individual in the offspring population is also a member of the parent population, and that the population transformation depends on the fitnesses of the individuals in the parent population. The following definition reflects these characteristics, in contrast to Bäck and Schwefel's, which admits any population transformation $s : (I^{\mu'} \cup I^{\mu'+\mu}) \longrightarrow I^{\mu}$.

Definition 2.3.10 (Selection operator): *Let $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I, \mathbb{R}), \Omega)$. If for every $P \in I^{\mu}$, every $\Theta \in \mathbb{X}$, and every fitness function $\Phi : I \longrightarrow \mathbb{R}$, s satisfies*

$$\mathbf{a} \in s_{(\Theta,\Phi)}(P) \Longrightarrow \mathbf{a} \in P \quad .$$

then s is called a selection operator. □

Bäck [6] formally defines specific probabilistic selection operators in terms of *selection probabilities.* The following definition is equivalent to his, but notationally reflects the usual dependence of selection probabilities on both the parent population and the fitness function.

Definition 2.3.11 (Selection probability): *Let $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I, \mathbb{R}), \Omega)$ be a selection operator, $\Theta \in \mathbb{X}$, $\Phi : I \longrightarrow \mathbb{R}$ a fitness function, $P \in I^{\mu}$, and $\mathbf{a} \in P$. Then*

$$p_{sel}(\mathbf{a}; s_{(\Theta,\Phi)}, P) \quad \triangleq \quad \Pr[\mathbf{a} \in s_{(\Theta,\Phi)}(P) \mid \mathbf{a} \in P]$$

is the selection probability *assigned to $\mathbf{a} \in P$ by $s_{(\Theta,\Phi)}$.* □

A selection operator is order-based if order preserving transformations of the fitness function also preserve selection probabilities of individuals.

Definition 2.3.12 (Order-based selection operator): *Let $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I, \mathbb{R}), \Omega)$ be a selection operator. If for every $\Theta \in \mathbb{X}$ (the operator parameters), every $D : I \longrightarrow \mathbb{R}^m$ (the decoding function), every $f : \mathbb{R}^m \longrightarrow \mathbb{R}$ (the objective function), every order-preserving fitness scaling function $T_s : \mathbb{R} \longrightarrow \mathbb{R}$, every population P, and every individual $\mathbf{a} \in P$, s satisfies*

$$p_{sel}(\mathbf{a}; s_{(\Theta, f \circ D)}, P) \quad = \quad p_{sel}(\mathbf{a}; s_{(\Theta, T_s \circ f \circ D)}, P) \quad ,$$

then s is called an order-based selection operator. □

All linkage-friendly genetic algorithms (defined in Section 2.6) use order-based selection operators. Research focuses on those which use tournament selection (defined in Section 2.6.2.3). Other examples of order-based selection operators include ranking selection, (μ, λ) selection, and $(\mu + \lambda)$ selection, which are discussed by Bäck [6].

2.3.3 Algorithmic Specification. The preceding definitions of the various types of evolutionary operators permit the following formal definition of an evolutionary algorithm, due essentially to Bäck and Schwefel.

Definition 2.3.13 (Evolutionary algorithm): *Let*

- *I be a non-empty set (the individual space),*
- *$\{\mu^{(i)}\}_{i \in \mathbb{N}}$ a sequence in $\mathbb{Z}^+$ (the parent population sizes),*
- *$\{\mu'^{(i)}\}_{i \in \mathbb{N}}$ a sequence in $\mathbb{Z}^+$ (the offspring population sizes),*
- *$\Phi : I \longrightarrow \mathbb{R}$ a fitness function,*
- *$\iota : \bigcup_{i=1}^{\infty} (I^{\mu})^i \longrightarrow \{\texttt{true,false}\}$ (the termination criterion),*
- *$\chi \in \{\texttt{true,false}\}$,*
- *r a sequence $\{r^{(i)}\}$ of recombination operators*

$$r^{(i)} : \mathbb{X}_r^{(i)} \longrightarrow \mathcal{T}\left(\Omega_r^{(i)}, \mathcal{T}\left(I^{\mu^{(i)}}, I^{\mu'^{(i)}}\right)\right)$$

- *m a sequence $\{m^{(i)}\}$ of mutation operators*

$$m^{(i)} : \mathbb{X}_m^{(i)} \longrightarrow \mathcal{T}\left(\Omega_m^{(i)}, \mathcal{T}\left(I^{\mu'^{(i)}}, I^{\mu'^{(i)}}\right)\right)$$

- *s a sequence $\{s^{(i)}\}$ of selection operators*

$$s^{(i)} : \mathbb{X}_s^{(i)} \times \mathcal{T}(I, \mathbb{R}) \longrightarrow \mathcal{T}\left(\Omega_s^{(i)}, \mathcal{T}\left(\left(I^{\mu'^{(i)}+\chi\mu^{(i)}}\right), I^{\mu^{(i+1)}}\right)\right)$$

- $\Theta_r^{(i)} \in \mathbb{X}_r^{(i)}$ *(the* recombination parameters*),*
- $\Theta_m^{(i)} \in \mathbb{X}_m^{(i)}$ *(the* mutation parameters*), and*
- $\theta_s^{(i)} \in \mathbb{X}_s^{(i)}$ *(the* selection parameters*).*

Then the algorithm shown in Figure 5 is called an evolutionary algorithm. □

```
$t := 0$:
initialize $P(0) := \{\mathbf{a}_1(0), \ldots, \mathbf{a}_\mu(0)\} \in I^{\mu^{(0)}}$;
while $(\iota(\{P(0), \ldots, P(t)\}) \neq \mathtt{true})$ do
        recombine: $P'(t) := r^{(t)}_{\Theta_r^{(t)}}(P(t))$;
        mutate: $P''(t) := m^{(t)}_{\Theta_m^{(t)}}(P'(t))$;
        select: if $\chi$
                then $P(t+1) := s^{(t)}_{(\theta_s^{(t)}, \Phi)}(P''(t))$;
                else $P(t+1) := s^{(t)}_{(\theta_s^{(t)}, \Phi)}(P''(t) \cup P(t))$;
              fi
        $t := t + 1$;
od
```

Figure 5. Outline of an Evolutionary Algorithm

This definition differs from Bäck and Schwefel's in several ways. First, and most importantly, the population sizes, operators, and parameters are all represented as sequences, reflecting the fact that certain evolutionary algorithms use varying population sizes, use multiple phases of execution in which different operators are applied, and vary their parameters over the course of execution. In particular, some linkage-friendly genetic algorithms (defined in Section 2.6) exhibit these characteristics.

Another difference between the definitions is that in Figure 5, the termination condition ι depends on the set of populations $\{P(0), \ldots, P(t)\}$. Many evolutionary algorithms terminate after a fixed number of generations (corresponding to a termination criterion satisfying $\iota(\{P(0), \ldots, P(t)\}) = \mathtt{true} \iff$

$\operatorname{card}(\{P(0), \ldots, P(t)\}) > t_f$), or based on conditions involving populations previous to the current generation. Both definitions fail to include evolutionary algorithms which terminate based on conditions involving the number of function evaluations performed.

Two further differences are notational. The variable χ is introduced to preserve Bäck and Schwefel's explicit representation of selection operators which act on populations of size $\mu' + \mu$, as well as those which act on populations of size μ'. In Bäck and Schwefel's definition, selection acts on the population $P''(t) \cup Q$, where $Q \in \{\{\}, P(t)\}$.

Finally, the fitness function is represented as a parameter of the selection operator. Consequently, explicit statement of the evaluation step is unnecessary.

To summarize, the concepts developed in this section include population transformations, random population transformations, and general evolutionary operators, as well as recombination, mutation, and selection operators. The development results in a general yet precise formal framework for the class of evolutionary algorithms. In later sections, specific algorithms are defined in the context of this framework.

2.4 Simple Genetic Algorithms

Much of the genetic algorithms literature relates to the simple genetic algorithm (sGA), defined by Goldberg [29] based on Holland's seminal work [43], or slight variations thereof. This section defines the sGA in the framework of evolutionary algorithms established in Section 2.3. Section 2.4.1 discusses the fixed-length binary string representation used by the sGA. The next section discusses the evolutionary operators of the sGA: single-point crossover (Section 2.4.2.1), point mutation (Section 2.4.2.2), and stochastic selection with replacement (Section 2.4.2.3). Finally, Section 2.4.3 presents the specification of the sGA.

2.4.1 Representation. Let $\mathcal{A}$ be a non-empty set (the *genic alphabet*), $\ell \in \mathbb{Z}^+$ (the *string length*), and $\mathcal{L} \triangleq \{1, \ldots, \ell\}$ (the *loci*). Then the *individual space* is $I \triangleq \mathcal{A}^\ell$, and an *individual* is a finite sequence

$\mathbf{a} = (a_1, \ldots, a_\ell) \in I$. Individuals are sometimes referred to as *chromosomes*[4]. Each $a_i \in \mathcal{A}$ is an *allele*, each $i \in \mathcal{L}$ is a *locus*, and each ordered pair (a_i, i) is a *gene*. For many genetic algorithms $\mathcal{A} = \{0, 1\}$, in which case the individuals are described as *binary*, and the algorithm is called *binary-coded*. The simple genetic algorithm is binary-coded.

2.4.2 Genetic Operators.

This section discusses the evolutionary operators used by the simple genetic algorithm. The single-point crossover operator is defined in Section 2.4.2.1, and Section 2.4.2.2 defines the point mutation operator. Finally, roulette wheel selection is defined in Section 2.4.2.3.

2.4.2.1 Recombination.

The recombination operators used in genetic algorithms are called *crossover operators*. They are traditionally viewed by genetic algorithm researchers as the primary mechanism by which new solutions are introduced to the genetic algorithm search process. Numerous crossover operators in use, including single-point crossover, two-point crossover, multi-point crossover, uniform crossover, and a host of domain specific crossover operators, especially in the context of combinatoric optimization problems.

The simple genetic algorithm uses single-point crossover, which is parameterized by the *probability of crossover* p_c. The individuals in the parent population are randomly paired, and a crossover point is randomly chosen for each pair. Those portions of the parent individuals following the crossover points are exchanged with probability p_c to form pairs of offspring individuals. In the following definition, and the sequel, the set of permutations on $\{1, \ldots, n\}$ is denoted π_n.

Definition 2.4.1 (Single-point crossover operator): *Let $\mathcal{A}$ be a non-empty set (the* genic alphabet*), $\ell \in \mathbb{Z}^+$ (the* individual length*), $I \triangleq \mathcal{A}^\ell$ (the* individual space*), $\mu = \mu' \in \mathbb{Z}^+$ (the* population size*), $\nu \triangleq \lfloor \frac{\mu}{2} \rfloor$, $\Omega \triangleq \pi_\mu \times [0, 1]^\nu \times \{1, \ldots, \ell - 1\}^\nu$, $\omega \triangleq (\sigma, \mathbf{X}, \mathbf{Y}) \sim U(\Omega)$, and $r : \mathbb{R} \longrightarrow \mathcal{T}(\Omega, \mathcal{T}(I^\mu, I^\mu))$ an evolutionary operator. If for every $p_c \in [0, 1]$ (the* probability of crossover*), $P \in I^\mu$, $i \in \{1, \ldots, \nu\}$, and $j \in \{1, \ldots, \ell\}$, r*

[4] A word on terminology is in order. In much of the GA literature, terms from evolution theory, biology, and genetics are used (abused?) freely to refer to the computational concepts which they inspire. This tradition encourages anthropomorphizing the algorithms, but in the interest of remaining consistent with the literature this research adopts the standard terminology.

satisfies

$$([r_{p_c}(P)]_{2i-1})_j = \begin{cases} [P_{\sigma(2i-1)}]_j & , \textit{if } X_i > p_c \textit{ or } j \leq Y_i \\ [P_{\sigma(2i)}]_j & , \textit{if } X_i \leq p_c \textit{ and } j > Y_i \end{cases} ,$$

$$([r_{p_c}(P)]_{2i})_j = \begin{cases} [P_{\sigma(2i)}]_j & , \textit{if } X_i > p_c \textit{ or } j \leq Y_i \\ [P_{\sigma(2i-1)}]_j & , \textit{if } X_i \leq p_c \textit{ and } j > Y_i \end{cases} ,$$

and

$$[r_{p_c}(P)]_\mu = P_{\sigma(\mu)} \quad , \textit{if } \mu \textit{ is odd} \quad ,$$

then r is called a single-point crossover operator. □

Single-point crossover is illustrated in Figure 6. Note that single-point crossover is restricted to parents of equal length.

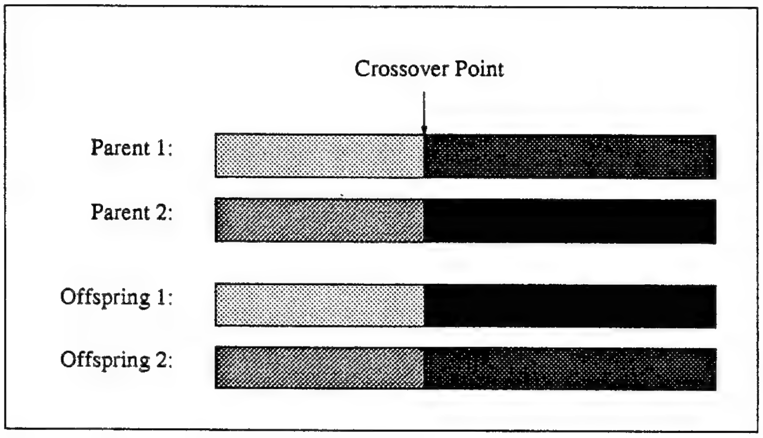


Figure 6. A single-point crossover operator acting on a parent population of size $\mu = 2$.

2.4.2.2 Mutation. Mutation is viewed as a "background" operator by many genetic algorithm researchers. which is very much in contrast to the view of the evolution strategies community. The simple genetic algorithm uses a mutation operator called *point mutation*, which is parameterized by the *probability* *of mutation* p_m.[5] Each allele of each individual in the population is mutated with independent probability p_m.

Definition 2.4.2 (Point mutation operator): *Let $\mathcal{A}$ be a non-empty set (the* genic alphabet*), $\ell \in \mathbb{Z}^+$ (the* individual length*). $I \triangleq \mathcal{A}^\ell$ (the* individual space*), $\mu = \mu' \in \mathbb{Z}^+$ (the* population size*), $\Omega \triangleq [0,1]^{\mu \times \ell} \times \mathcal{A}^{\mu \times \ell}$, $\omega \triangleq (\underline{\mathbf{X}}, \underline{\mathbf{Y}}) \sim U(\Omega)$, and $m : \mathbb{R} \longrightarrow \mathcal{T}\left(\Omega, \mathcal{T}\left(I^{\mu'}, I^{\mu'}\right)\right)$ an evolutionary operator. If for every $p_m \in [0,1]$ (the* probability of mutation*), $P \in I^{\mu'}$, $i \in \{1, \ldots, \mu'\}$, and $j \in \{1, \ldots, \ell\}$, m satisfies*

$$([m_{p_m}(P)]_i)_j = \begin{cases} Y_{ij} & , \textit{if } X_{ij} \leq p_m, \textit{ and} \\ (P_i)_j & , \textit{if } X_{ij} > p_m \end{cases},$$

then m is called a point mutation operator. □

Point mutation is illustrated in Figure 7.

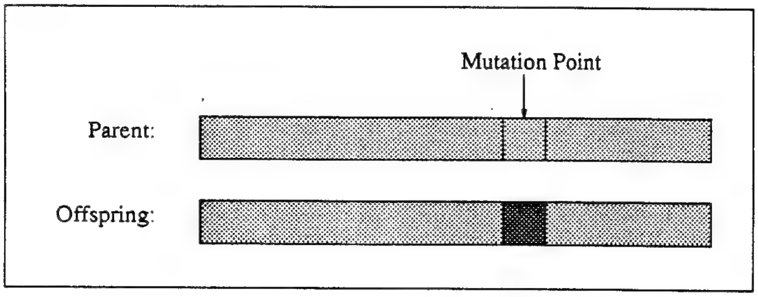


Figure 7. A point mutation operator acting on a population containing a single individual (the parent). A single mutation point is depicted.

2.4.2.3 Selection. Holland's original description of the genetic algorithm, on which the simple genetic algorithm is based, specifies that each individual's selection probability is proportional to its fitness. Selection operators which exhibit this characteristic are referred to as *fitness proportionate*. Fitness

[5] There is some ambiguity regarding the term "probability of mutation." In one convention, the allele of the parent individual is included in the set from which the allele of the offspring individual is drawn. For this convention, a probability of mutation $p_m = 1.0$ specifies random search, regardless of the cardinality of the genic alphabet. In the other convention, the allele of the parent individual is excluded. For this convention $p_m = \frac{C-1}{C}$ specifies random search, where C is the cardinality of the genic alphabet (which is assumed to be finite, else the point is moot). For convenience, the definition adopted in this research follows the former convention. This is in contrast to the convention followed in the specification of the simple genetic algorithm [29].

proportionate selection operators have no parameters other than the fitness function, and require that each individual have positive fitness.

Definition 2.4.3 (Fitness proportionate selection operator): *Let $s \in \mathcal{EVOP}(I, \mu, \mathcal{T}(I, \mathbb{R}^+), \Omega)$ be a selection operator. If for every $P \in I^\mu$, and fitness function $\Phi : I \longrightarrow \mathbb{R}^+$, the selection probability of each individual $\mathbf{a} \in P$ satisfies*

$$p_{sel}(\mathbf{a}; s_\Phi, P) \quad = \quad \frac{\Phi(\mathbf{a})}{\sum_{i=1}^{\mu} \Phi(P_i)} \quad .$$

then s is called a fitness proportionate selection operator. □

The simple genetic algorithm uses a fitness proportionate selection operator called *stochastic sampling with replacement*, also known as *roulette wheel selection* [29]. The latter term is motivated by imagining that each individual is assigned an arc on the perimeter of a roulette wheel, the length of which is proportional to the individual's fitness. Members of the offspring population are selected by "spinning the wheel," and including in the offspring population a copy of the individual within whose arc the roulette ball lands. The spins of the wheel correspond to the components of the random vector ω in the following definition.

Definition 2.4.4 (Stochastic selection with replacement operator): *Let $\Omega \triangleq [0, 1]^\mu$, $\omega \sim U(\Omega)$, $s \in \mathcal{EVOP}(I, \mu, \mathcal{T}(I, \mathbb{R}^+), \Omega)$, and*

$$\sigma(k; s, \Phi, P) \quad \triangleq \quad \min\left\{ j : \sum_{i=1}^{j} p_{sel}(P_j; s_\Phi, P) \geq \omega_k \right\} \quad .$$

If for every fitness function $\Phi : I \longrightarrow \mathbb{R}^+$ and every population $P \in I^\mu$, s satisfies

$$[s(P)]_i \quad = \quad P_{\sigma(i; s, \Phi, P)} \quad .$$

then s is called a stochastic selection with replacement operator. □

2.4.3 Algorithmic Specification. The preceding sections describe the individual space and each of the evolutionary operators of a simple genetic algorithm. This section specifies the simple genetic algorithm in the formal framework developed in Section 2.3.

Definition 2.4.5 (Simple genetic algorithm): *Let*

- $\ell \in \mathbb{Z}^+$ *(the individual length).*
- $I \triangleq \{0, 1\}^\ell$ *(the individual space).*
- $t_f \in \mathbb{Z}^+$ *(the final generation).*
- $\mu = \mu' \in \mathbb{Z}^+$ *(the population size).*
- $\Phi : I \longrightarrow \mathbb{R}$ *a fitness function.*
- $\iota : \bigcup_{i=1}^{\infty} (I^\mu)^i \longrightarrow \{\texttt{true},\texttt{false}\}$ *(the termination criterion) such that*

$$\iota(\{P(0), \ldots, P(t)\}) = \texttt{true} \Longleftrightarrow \mathit{card}(\{P(0), \ldots, P(t)\}) > t_f \quad .$$

- $r \in \mathcal{EVOP}(I, \mu, \mathbb{R}, \Omega_r)$ *a single-point crossover operator,*
- $m \in \mathcal{EVOP}(I, \mu, 1, \Omega_m)$ *a point mutation operator,*
- $s : \mathcal{T}(I, \mathbb{R}^+) \longrightarrow \mathcal{T}(\Omega, \mathcal{T}(I^\mu, I^\mu))$ *a stochastic selection with replacement operator. and*
- $\Theta_r, \Theta_m \in \mathbb{R}$.

Then the algorithm shown in Figure 8 is called a simple genetic algorithm. □

Although the simple genetic algorithm is in widespread use as an optimum seeking technique, it suffers from at least two significant disadvantages in this application compared to other evolutionary algorithms. One drawback is that its effectiveness with respect to a given application depends on the decoding function. In particular, the effectiveness typically depends on the "order" in which the genes are mapped to the object variables of the objective function. Because the order is fixed, the simple genetic algorithm possesses no

```
$t := 0$:
initialize $P(0) := \{\mathbf{a}_1(0), \ldots, \mathbf{a}_\mu(0)\} \in I^\mu$;
while $(\iota(\{P(0), \ldots, P(t)\}) \neq$ true) do
        recombine: $P'(t) := r_{\Theta_r}(P(t))$:
        mutate: $P''(t) := m_{\Theta_m}(P'(t))$;
        select: $P(t+1) := s_\Phi(P''(t))$;
        $t := t + 1$:
od
```

Figure 8. Outline of a Simple Genetic Algorithm

mechanism by which to detect "linkage" between strongly interacting genes and adapt the representation scheme accordingly.

Another limitation of the simple genetic algorithm is that its effectiveness also depends on the fitness scaling function. This dependence is directly attributable to the use of fitness proportionate selection. This relationship is addressed in more detail in Section 2.5, where it is also shown that algorithms using order-based selection operators do not share this disadvantage.

2.5 Invariance Properties of Selection Operators

The set of evolutionary operators employed by an evolutionary algorithm determines the effectiveness of the algorithm for a given application. This section identifies several properties which characterize certain selection operators. and which relate to the effects those operators have on an algorithm's effectiveness and efficiency. Most importantly, the class of strictly invariant selection operators is defined and shown to be equivalent to the class of order-based selection operators. All linkage-friendly genetic algorithms (defined in Section 2.6) use selection operators of this type.

If two functions $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $\hat{f} : \mathbb{R}^n \longrightarrow \mathbb{R}$ are related by $\hat{f}(\cdot) = af(\cdot) + b$ where $a, b \in \mathbb{R}$ and $a > 0$, then f and $\hat{f}$ share the same (local and global) maxima and minima, as well as other important properties. Intuitively. a desirable characteristic for an optimum seeking technique is that it be equally effective with respect to such functions. This characteristic is closely related to the selection operator properties of *scale invariance* and *translation invariance*. These properties are defined by de la Maza and Tidor [15], although their implications for commonly used genetic algorithm selection operators are well understood in earlier studies (see Grefenstette and Baker [40], for example).

The following definitions are equivalent to those proposed by de la Maza and Tidor. A selection operator is scale invariant if the selection probabilities which it assigns are preserved when the fitness function is multiplied by a positive scalar.

Definition 2.5.1 (Scale invariant selection operator): *Let $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I, \mathbb{R}), \Omega)$ be a selection operator. If for every $\Theta \in \mathbb{X}$, every fitness function $\Phi : I \longrightarrow \mathbb{R}$, every population $P \in I^{\mu}$, every individual $\mathbf{a} \in P$, and every $c \in \mathbb{R}^{+}$*

$$p_{sel}(\mathbf{a}; s_{(\Theta, \Phi)}, P) \quad = \quad p_{sel}(\mathbf{a}; s_{(\Theta, c\Phi)}, P) \quad ,$$

then s is called a scale invariant selection operator. □

All selection operators in common use are scale invariant. In contrast, some commonly used selection operators, including all fitness proportionate operators, are not translation invariant. A selection operator is translation invariant if the selection probabilities which it assigns are preserved when a constant (function) is added to the fitness function.

Definition 2.5.2 (Translation invariant selection operator): *Let $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I, \mathbb{R}), \Omega)$ be a selection operator, and $u : I \longrightarrow \mathbb{R}$ such that $u(\mathbf{a}) \triangleq 1$ for every $\mathbf{a} \in I$. If for every $\Theta \in \mathbb{X}$, every fitness function $\Phi : I \longrightarrow \mathbb{R}$, every population $P \in I^{\mu}$, every individual $\mathbf{a} \in P$, and every $c \in \mathbb{R}$*

$$p_{sel}(\mathbf{a}; s_{(\Theta, \Phi)}, P) \quad = \quad p_{sel}(\mathbf{a}; s_{(\Theta, \Phi + cu)}, P) \quad .$$

then s is called a translation invariant selection operator. □

The use of selection operators which are not translation invariant, including those which are fitness proportionate, leads some researchers to develop a large body of empirical knowledge regarding appropriate fitness scaling functions for various applications (see Michalewicz [57], for example).

In contrast, this research is primarily concerned with selection operators which are translation invariant. More specifically, it is concerned with the class of selection operators which are invariant under every strictly increasing transformation.

Definition 2.5.3 (Strictly invariant selection operator): *Let $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I, \mathbb{R}), \Omega)$ be a selection operator. If for every $\Theta \in \mathbb{X}$, every fitness function $\Phi : I \longrightarrow \mathbb{R}$, every population $P \in I^{\mu}$, every individual $\mathbf{a} \in P$, and every strictly increasing function $g : \mathbb{R} \longrightarrow \mathbb{R}$*

$$p_{sel}(\mathbf{a}; s_{(\Theta,\Phi)}, P) \quad = \quad p_{sel}(\mathbf{a}; s_{(\Theta, g \circ \Phi)}, P) \quad .$$

then s is called a strictly invariant selection operator. □

Because functions of the form $f(x) \overset{\triangle}{=} ax + b$ are strictly increasing when $a > 0$, strict invariance implies both scale and translation invariance. This is stated formally in the following theorem:

Theorem 2.5.4 *Let s be a strictly invariant selection operator. Then s is scale invariant and translation invariant.*

Proof: By the definition of a selection operator, $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I, \mathbb{R}), \Omega)$ for some non-empty set I, $\mu \in \mathbb{Z}^{+}$, set $\mathbb{X}$ (the *parameter space*), and set Ω (the *sample space*). Let $\Theta \in \mathbb{X}$, $\Phi : I \longrightarrow \mathbb{R}$, $P \in I^{\mu}$, and $\mathbf{a} \in P$.

Let $c \in \mathbb{R}^{+}$ and define $g : \mathbb{R} \longrightarrow \mathbb{R}$ by $g(x) \overset{\triangle}{=} cx$. Then g is strictly increasing. Because s is strictly invariant, $p_{sel}(\mathbf{a}; s_{(\Theta,\Phi)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,g\circ\Phi)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,c\Phi)}, P)$. Because Θ, Φ, P, $\mathbf{a}$, and c are arbitrary, s is scale invariant.

Let $c \in \mathbb{R}$ and define $g : \mathbb{R} \longrightarrow \mathbb{R}$ by $g(x) \triangleq x + c$, and $u : I \longrightarrow \mathbb{R}$ such that $u(\mathbf{a}) \triangleq 1$ for every $\mathbf{a} \in I$. Then g is strictly increasing, so that $p_{sel}(\mathbf{a}; s_{(\Theta,\Phi)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,g\circ\Phi)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,\Phi+cu)}, P)$. Because Θ, Φ, P, $\mathbf{a}$, and c are arbitrary, s is translation invariant. ■

The following theorem states that strictly invariant selection operators necessarily assign selection probabilities based solely on the (possibly partial) ordering induced on the individual space by the fitness function, and that all selection operators which assign selection probabilities in such a manner are strictly invariant.

Theorem 2.5.5 *A selection operator is strictly invariant if and only if it is order-based.*

Proof: Let s be a selection operator. Then by the definition of a selection operator, $s \in \mathcal{EVOP}(I, \mu, \mathbb{X} \times \mathcal{T}(I,\mathbb{R}), \Omega)$ for some non-empty set I, $\mu \in \mathbb{Z}^+$, set $\mathbb{X}$ (the *parameter space*), and set Ω (the *sample space*).

"If": Suppose s is an order-based selection operator. Let $\Theta \in \mathbb{X}$, $\Phi = T_s \circ f \circ D : I \longrightarrow \mathbb{R}$ a fitness function, $P \in I^\mu$, $\mathbf{a} \in P$, and $g : \mathbb{R} \longrightarrow \mathbb{R}$ strictly increasing. Define $\hat{f} \triangleq T_s \circ f$. Then $\Phi = \hat{f} \circ D$. Also, $D : I \longrightarrow \mathbb{R}^n$ and $\hat{f} : \mathbb{R}^n \longrightarrow \mathbb{R}$ for some $n \in \mathbb{N}$. Furthermore, by Lemma 2.3.4, g is an order-preserving fitness scaling function. Thus, by the definition of an order-based selection operator, $p_{sel}(\mathbf{a}; s_{(\Theta,\Phi)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,\hat{f}\circ D)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,g\circ\hat{f}\circ D)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,g\circ\Phi)}, P)$. Because Θ, Φ, P, $\mathbf{a}$, and g are arbitrary, s is strictly invariant.

"Only if": Suppose that s is strictly invariant. Let $\Theta \in \mathbb{X}$, $D : I \longrightarrow \mathbb{R}^n$ for some $n \in \mathbb{N}$, $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $T_s : \mathbb{R} \longrightarrow \mathbb{R}$ an order-preserving fitness scaling function, $P \in I^\mu$, and $\mathbf{a} \in P$. Also, let $g : \mathbb{R} \longrightarrow \mathbb{R}$ be the identity mapping, and define $\Phi \triangleq g \circ f \circ D$. Then $f \circ D = g \circ f \circ D = \Phi$. Also, by Lemma 2.3.4, T_s is strictly increasing. Thus, by the definition of a strictly invariant selection operator, $p_{sel}(\mathbf{a}; s_{(\Theta,f\circ D)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,\Phi)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,T_s\circ\Phi)}, P) = p_{sel}(\mathbf{a}; s_{(\Theta,T_s\circ f\circ D)}, P)$. Because Θ, D, f, T_s, P, and $\mathbf{a}$ are arbitrary, s is order-based. ■

In light of this theorem, it is not surprising that in practice algorithms which use order-based selection operators rarely use (nontrivial) fitness scaling functions. This section concludes with the observation that order-based selection operators are necessarily scale and translation invariant.

Corollary 2.5.6 *Let s be an order-based selection operator. Then s is scale invariant and translation invariant.*

Proof: By Theorem 2.5.5, s is strictly invariant. By Theorem 2.5.4, s is scale invariant and translation invariant. ■

2.6 Linkage-Friendly Genetic Algorithms

The effectiveness of the simple genetic algorithm with respect to a given application depends on the specified decoding function. In particular, the effectiveness depends on the "order" in which the genes are mapped to the object variables.[6] The effectiveness also depends on the specified fitness scaling function. These dependencies lead researchers to consider another class of evolutionary algorithms, which lack these dependencies. In this research, these algorithms are collectively called *linkage-friendly genetic algorithms (lfGAs).*[7]

Historically, the dependence of the simple genetic algorithm's effectiveness on the decoding function motivated Goldberg, et al. [36] to propose the *messy genetic algorithm (mGA).* Later, efficiency considerations motivated the development of the *fast messy genetic algorithm (fmGA)* [35]. More recently, Kargupta extended the fmGA to give explicit consideration to the equivalence class competitions conducted, resulting in the *gene expression messy genetic algorithm* (gemGA) [45]. The representation scheme shared by the mGA and fmGA (Section 2.6.1), as well as the representation scheme of the gemGA, is such that the effectiveness of each algorithm is independent of the "order" in which genes are mapped to object variables.

[6] This fact follows immediately from Holland's Schema Theorem [43].

[7] The term "linkage friendly genetic algorithms" is due to Goldberg [30].

The recombination, mutation, and selection operators used by the mGA and fmGA are discussed in Section 2.6.2. The selection operator used by both algorithms (and by the gemGA) is such that the effectiveness of each is independent of the fitness scaling function. The general evolutionary algorithm framework developed in Section 2.3 is used to formally specify the mGA and fmGA (Sections 2.6.3 and 2.6.4, respectively). The section concludes with a review of existing fmGA parameter selection techniques (Section 2.6.5).

2.6.1 Representation. Linkage-friendly genetic algorithms as defined in this research share a common representation scheme. In contrast to the representation scheme used in simple genetic algorithms, loci are represented explicitly and individuals are not necessarily of uniform length.

Definition 2.6.1 (Linkage-friendly genetic algorithm (lfGA) individual space): *Let $\mathcal{A}$ be a non-empty set (the* genic alphabet*), $\ell \in \mathbb{Z}^+$ (the* nominal string length*). $\mathcal{L} \triangleq \{1, \ldots, \ell\}$ (the* loci*), and $o \in \mathbb{R}$ such that $o \geq 1$ (the* overflow factor*). Then*

$$I \triangleq \bigcup_{\lambda=0}^{\lfloor o \cdot \ell \rfloor} (\mathcal{A} \times \mathcal{L})^\lambda \simeq \bigcup_{\lambda=0}^{\lfloor o \cdot \ell \rfloor} (\mathcal{A}^\lambda \times \mathcal{L}^\lambda)$$

is called an lfGA individual space over $\mathcal{A}$.

□

Each $a_i \in \mathcal{A}$ is an *allele*, each $l_i \in \mathcal{L}$ is a *locus* (plural *loci*), and each ordered pair (a_i, l_i) is a *gene* (c.f. Section 2.4.1). Thus, an *lfGA individual* $\mathbf{x} \in I$ may be viewed as a vector $((a_1, l_1), \ldots, (a_\lambda, l_\lambda))$ of allele-locus pairs for some $\lambda \in \{0, \ldots, \lfloor o \cdot \ell \rfloor\}$ (the *string length* or *individual length*). Alternatively, an lfGA individual may be viewed as an ordered pair of equal length vectors $\mathbf{x} = (\mathbf{a}, \mathbf{l}) = ((a_1, \ldots, a_\lambda), (l_1, \ldots, l_\lambda)) \in \mathcal{A}^\lambda \times \mathcal{L}^\lambda$. This research uses the two views interchangeably as convenient.

Given an individual $(\mathbf{a}, \mathbf{l}) = ((a_1, \ldots, a_\lambda), (l_1, \ldots, l_\lambda))$, a locus L may occur zero, one, or more times in $\mathbf{l}$. This implies that individuals need not completely specify a candidate solution, and also that individuals may overspecify components of candidate solutions. In *non-overspecified individuals* each locus occurs no more than once (i.e. $l_i = l_j \iff i = j$), hence such individuals have lengths $\lambda \in \{0, \ldots, \ell\}$. It is convenient to define the set of length λ non-overspecified individuals

$$I(\lambda) \quad \triangleq \quad \{(\mathbf{a},\mathbf{l}) = ((a_1,\ldots,a_\lambda),(l_1,\ldots,l_\lambda)) \in I : l_i = l_j \Longleftrightarrow i = j\} \quad . \tag{2}$$

An individual $(\mathbf{a},\mathbf{l})$ is *fully specified* if each locus occurs exactly once, i.e. if $(\forall i \in \mathcal{L})(\exists! j \in \mathcal{L})[l_j = i]$. The set of fully specified individuals is thus $I_F \triangleq I(\ell)$. There is of course a "natural decoding" $\Gamma_F : I_F \longrightarrow \mathcal{A}^\ell$ which, given a fully specified individual, produces an ℓ-vector of alleles representing a candidate solution.[8] More generally, given a fully specified individual $\mathbf{c} \in I_F$, referred to as a *competitive template*, the overlay mapping associates every individual $\mathbf{x} \in I$ (fully specified or otherwise) with an ℓ-vector of alleles.

Definition 2.6.2 (Overlay mapping): *Let I be an lfGA individual space over the genic alphabet $\mathcal{A}$ with nominal string length ℓ, and $I_F \triangleq I(\ell)$ defined by Equation 2. The mapping $\Gamma : I \times I_F \longrightarrow \mathcal{A}^\ell$ such that for each $i \in \{1,\ldots,\ell\}$*

$$[\Gamma((\mathbf{a},\mathbf{l}),(\mathbf{b},\mathbf{m}))]_i \quad \triangleq \quad \begin{cases} a_j, & \textit{if } \; j \triangleq \min\{k : l_k = i\} \;\; \textit{exists} \\ b_j \;\; \textit{where} \;\; m_j = i, & \textit{if } \; \forall k : l_k \neq i \end{cases}$$

is called the overlay mapping *for I.* □

The association of each individual $\mathbf{x} \in I$ with a vector of alleles via the overlay mapping may be thought of as the first step in assigning a fitness to $\mathbf{x}$. Subsequent steps include mapping the vector of alleles to the parameter space of the objective function, evaluation of the objective function, and possibly fitness scaling. The composition of these mappings is the lfGA fitness function.

Definition 2.6.3 (Linkage-friendly genetic algorithm (lfGA) fitness function): *Let I be an lfGA individual space over the genic alphabet $\mathcal{A}$ with nominal string length ℓ, $I_F \triangleq I(\ell)$ defined by Equation 2, $\Gamma : I \times I_F \longrightarrow \mathcal{A}^\ell$ the overlay mapping for I, $D : \mathcal{A}^\ell \stackrel{\cdot}{\longrightarrow} \mathbb{R}^n$ (the* lfGA decoding function*), $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ (the* objective function*), $T_s : \mathbb{R} \longrightarrow \mathbb{R}$ (the* fitness scaling function*), and $\Phi \triangleq T_s \circ f \circ D \circ \Gamma : I \times I_F \longrightarrow \mathbb{R}$. Then*

[8] In particular, define $\Gamma_F : I_F \longrightarrow \mathcal{A}^\ell$ such that $[\Gamma(\mathbf{a},\mathbf{l})]_i \triangleq a_j$, where $l_j = i$.

$\Phi(\mathbf{x}, \mathbf{c})$ *denotes the* fitness of $\mathbf{x} \in I$ with respect to $\mathbf{c} \in I_F$. *Furthermore, given* $\mathbf{c} \in I_F$ *define* $\Phi_{\mathbf{c}} : I \longrightarrow \mathbb{R}$ *by* $\Phi_{\mathbf{c}}(\cdot) \triangleq \Phi(\cdot, \mathbf{c})$. *Then* $\Phi_{\mathbf{c}}$ *is called an* lfGA fitness function *for* I. □

Of course, an lfGA fitness function $\Phi_{\mathbf{c}}$ may be written as the composition $T_s \circ f \circ D_{\mathbf{c}} : I \longrightarrow \mathbb{R}$, where $D_{\mathbf{c}}(\cdot) \triangleq D(\Gamma(\cdot, \mathbf{c}))$. Thus, lfGA fitness functions are fitness functions in the sense of Definition 2.3.2. Finally, description of specific linkage-friendly genetic algorithms is considerably simplified by the following definition.

Definition 2.6.4 ((Order-k) potential building block): *Let* $\mathcal{A}$ *be a non-empty set (the* genic alphabet*),* $\ell \in \mathbb{Z}^+$ *(the* nominal string length*),* $\mathcal{L} \triangleq \{1, \ldots, \ell\}$ *(the* loci*), and* $\mathcal{S} \triangleq \{(a_1, l_1), \ldots, (a_k, l_k)\} \in 2^{\mathcal{A} \times \mathcal{L}}$ *a set of genes. If the loci of* $\mathcal{S}$ *are distinct, i.e.* $\mathcal{S}$ *satisfies* $i = j \Longleftrightarrow l_i = l_j$, *then it is called an* order-k potential building block *or simply a* potential building block. □

2.6.2 Genetic Operators. This section discusses the recombination, mutation, and selection operators used by the messy genetic algorithm (mGA) and fast messy genetic algorithm (fmGA). Both the mGA and the fmGA process individuals of non-uniform length, and consequently require a more general recombination operator than single-point crossover. The recombination operator proposed by Goldberg, et al. [36] is called the *cut-and-splice operator* (Section 2.6.2.1).

This research does not formally define the mGA mutation operator, which is analogous to the point mutation operator of the simple genetic algorithm, because all reported mGA experiments use a zero probability of mutation. In contrast, the fmGA uses a *building block filtering* operator (Section 2.6.2.2) which this research views as a mutation operator.

Finally, Section 2.6.2.3 formally defines the *binary tournament selection with thresholding* operator, which is order-based and used by both the mGA and the fmGA. Because it is order-based, the effectiveness of each algorithm is independent of the fitness scaling function.

2.6.2.1 Recombination. The individual spaces of linkage-friendly genetic algorithms consist of individuals of non-uniform length (see Definition 2.6.1). Thus, the single-point crossover operator used in the simple genetic algorithm is not directly applicable in linkage-friendly genetic algorithms. The *cut-*

and-splice operator is a recombination operator which processes individuals of non-uniform length and is otherwise similar to single-point crossover. It is convenient to define the cut-and-splice operator in terms of the composition of distinct *cut* and *splice* operators.

A cut operator maps pairs of individuals (the *parents*) to 4-tuples of individuals (the *fragments*). For $\mathbf{a} = (a_1, \ldots, a_\lambda) \in (\mathcal{A} \times \mathcal{L})^\lambda$, the following definition denotes by $\mathbf{a}_{i:j}$ the fragment $(a_i, \ldots, a_j) \in (\mathcal{A} \times \mathcal{L})^{j-i+1}$, where $1 \leq i \leq j \leq \lambda$. Some fragments may be trivial, i.e. of length 0: these are denoted $\{\}$.[9]

Definition 2.6.5 (Cut operator): *Let I be an lfGA individual space, $\Omega \triangleq [0,1]^4$. $\omega \triangleq (X_a, X_b, \hat{Y}_a, \hat{Y}_b) \sim U(\Omega)$, and $\kappa : \mathbb{R} \longrightarrow \mathcal{T}(\Omega, \mathcal{T}(I^2, I^4))$ an evolutionary operator. If for every $p_c \in [0,1]$ (the* cut probability*), every $(\mathbf{a}, \mathbf{b}) \in (\mathcal{A} \times \mathcal{L})^{\lambda_a} \times (\mathcal{A} \times \mathcal{L})^{\lambda_b} \subseteq I^2$ (the* parents*), $Y_a \triangleq \lceil (\lambda_a - 1) \cdot \hat{Y}_a \rceil$ and $Y_b \triangleq \lceil (\lambda_b - 1) \cdot \hat{Y}_b \rceil$ (the* cut points*), κ satisfies*

$$\kappa_{p_c}(\mathbf{a}, \mathbf{b}) \triangleq \begin{cases} (\mathbf{a}_{1:Y_a}, \mathbf{a}_{Y_a+1:\lambda_a}, \mathbf{b}_{1:Y_b}, \mathbf{b}_{Y_b+1:\lambda_b}) & \text{, } \textit{if } \lambda_a\lambda_b > 0,\ X_a \leq p_c, \textit{ and } X_b \leq p_c \\ (\mathbf{a}_{1:Y_a}, \mathbf{a}_{Y_a+1:\lambda_a}, \mathbf{b}, \{\}) & \text{, } \textit{if } \lambda_a > 0,\ X_a \leq p_c, \textit{ and} \\ & \quad \textit{either } \lambda_b = 0 \textit{ or } X_b > p_c \\ (\mathbf{a}, \mathbf{b}_{1:Y_b}, \mathbf{b}_{Y_b+1:\lambda_b}, \{\}) & \text{, } \textit{if } \lambda_b > 0,\ X_b \leq p_c, \textit{ and} \\ & \quad \textit{either } \lambda_a = 0 \textit{ or } X_a > p_c \\ (\mathbf{a}, \mathbf{b}, \{\}, \{\}) & \text{, } \textit{if either } \lambda_a = 0 \textit{ or } X_a > p_c, \textit{ and} \\ & \quad \textit{either } \lambda_b = 0 \textit{ or } X_b > p_c \end{cases},$$

then κ is called a cut operator. □

A splice operator maps 4-tuples of individuals (the *fragments*) to n-tuples of individuals (the *offspring*), where $n \in \{2, 3, 4\}$. In the following definition, if $\mathbf{a} = (a_1, \ldots, a_{\lambda_a}) \in (\mathcal{A} \times \mathcal{L})^{\lambda_a}$ and $\mathbf{b} = (b_1, \ldots, b_{\lambda_b}) \in (\mathcal{A} \times \mathcal{L})^{\lambda_b}$ are fragments, then the offspring $(a_1, \ldots, a_{\lambda_a}, b_1, \ldots, b_{\lambda_b})$ is denoted $\mathbf{ab}$.

[9] This notation is consistent with the view of an lfGA individual $\mathbf{x} \in (\mathcal{A} \times \mathcal{L})^n$ as equivalent to the sequence of n allele-locus pairs which it implicitly defines. By the definition of a sequence (see Apostol [3]), $\mathbf{x}$ is then a function, i.e. a set of ordered pairs, $\{(1, x_1), \ldots, (n, x_n)\}$, where each $x_i \in \mathcal{A} \times \mathcal{L}$ is an allele-locus pair. Suppose $n = 0$. Then, $\mathbf{x}$ is the empty set of ordered pairs.

Definition 2.6.6 (Splice operator): *Let I be an lfGA individual space.* $\Omega \triangleq [0,1]^3$, $\omega \triangleq (X_{ab}, X_{bc}, X_{cd}) \sim U(\Omega)$, *and* $\zeta : \mathbb{R} \longrightarrow \mathcal{T}(\Omega, \mathcal{T}(I^4, I^2 \cup I^3 \cup I^4))$ *an evolutionary operator. If for every* $p_s \in [0,1]$ *(the* splice probability*), and every* $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \in I^4$ *(the* fragments*).* ζ *satisfies*

$$\zeta_{p_s}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \triangleq \begin{cases} (\mathbf{ab}, \mathbf{cd}) & , \textit{if } X_{ab} \le p_s \textit{ and } X_{cd} \le p_s \\ (\mathbf{ab}, \mathbf{c}, \mathbf{d}) & , \textit{if } X_{ab} \le p_s \textit{ and } X_{cd} > p_s \\ (\mathbf{a}, \mathbf{bc}, \mathbf{d}) & , \textit{if } X_{ab} > p_s \textit{ and } X_{bc} \le p_s \\ (\mathbf{a}, \mathbf{b}, \mathbf{cd}) & , \textit{if } X_{ab} > p_s, X_{bc} > p_s, \textit{ and } X_{cd} \le p_s \\ (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) & , \textit{if } X_{ab} > p_s, X_{bc} > p_s, \textit{ and } X_{cd} > p_s \end{cases} ,$$

then ζ *is called a* splice operator. □

A *local cut-and-splice operator*[10] is an evolutionary operator which produces population transformations expressible as the composition of the population transformations resulting from a cut operator, a permutation of the resulting fragments (possibly depending on the parameters and random events of the cut operator), and a splice operator.

Definition 2.6.7 (Local cut-and-splice operator): *Let I be an lfGA individual space.* $\Omega \triangleq [0,1]^4 \times [0,1]^3$, $\omega \triangleq (\omega_c, \omega_s) \sim U(\Omega)$, κ *a cut operator,* $\sigma : \mathbb{R} \times [0,1]^4 \longrightarrow \pi_4$, ζ *a splice operator, and* $r' : \mathbb{R}^2 \longrightarrow \mathcal{T}(\Omega, \mathcal{T}(I^2, I^2 \cup I^3 \cup I^4))$ *an evolutionary operator. If* r' *satisfies*

$$[r'_{(p_c, p_s)}(\omega)](\mathbf{a}, \mathbf{b}) = [\zeta_{p_s}(\omega_s)] \left(([\kappa_{p_c}(\omega_c)](\mathbf{a}, \mathbf{b}))_{[\sigma(p_c, \omega_c)](1)}, \ldots, ([\kappa_{p_c}(\omega_c)](\mathbf{a}, \mathbf{b}))_{[\sigma(p_c, \omega_c)](4)} \right) ,$$

then r' *is called a* local cut-and-splice operator. □

[10] With respect to both recombination and mutation operators, Bäck and Schwefel [7] distinguish between "macro-operators" (equivalent to the "population transformations" defined in this research) and "local operators," which map populations to individuals. Informally speaking, local operators capture the low-level, essential behavior of the corresponding macro-operators. Consequently, specific recombination and mutation operators are often defined in terms of local operators.

Strictly speaking, the local cut-and-splice operator defined in this research is not a local operator in the sense of Bäck and Schwefel, because it produces more than one individual.

The permutation mapping σ in Definition 2.6.7 is arbitrary. Different mappings correspond to different local cut-and-splice operators and result in different sets of potential offspring. Goldberg, et al. [36] propose a local cut-and-splice operator, for which the potential sets of nontrivial offspring[11] are illustrated in Figure 9. The permutation mapping of Goldberg's local cut-and-splice operator is intended to closely resemble the behavior of single-point crossover for individuals of length close to the nominal string length.

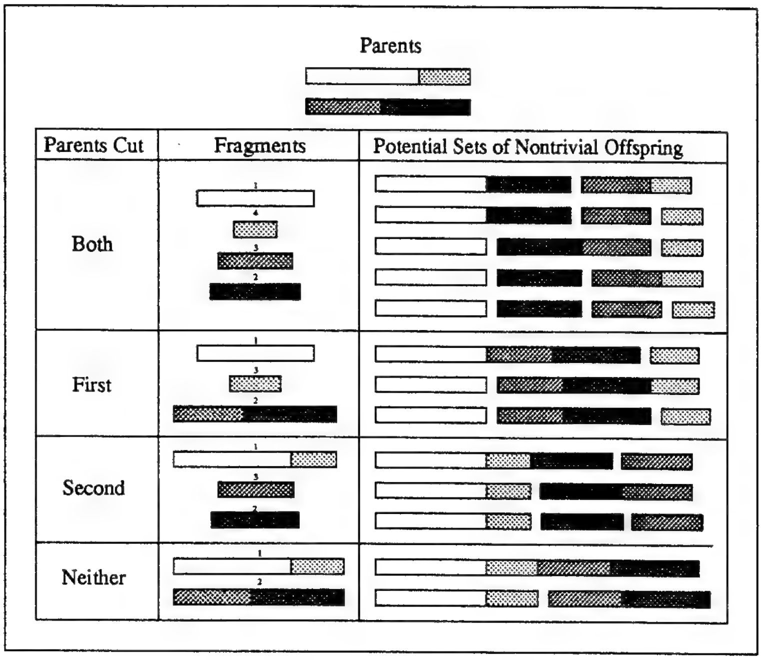


Figure 9. Potential Nontrivial Offspring Resulting From Goldberg's Local Cut-and-splice Operator

Definition 2.6.8 (Goldberg's local cut-and-splice operator): *Define* $\sigma : \mathbb{R} \times [0,1]^4 \longrightarrow \pi_4$ *by*

$$\sigma(p_c, \omega_c) \triangleq \begin{cases} (1,4,3,2) & , \textit{if } X_a \leq p_c \textit{ and } X_b \leq p_c \\ (1,2,3,4) & , \textit{if } X_a > p_c \textit{ and } X_b > p_c \\ (1,3,2,4) & , \textit{otherwise} \end{cases} .$$

Let I, Ω, ω, κ, ζ, and r' be as in Definition 2.6.7. Then r' is called Goldberg's local cut-and-splice operator.

□

[11] In practice, only nontrivial individuals are included in the offspring population.

A *cut-and-splice operator* is an evolutionary operator which extends a local cut-and-splice operator to operate on populations of arbitrary size (i.e. a macro-operator corresponding to a local cut-and-splice operator). In contrast to the situation with single-point crossover, for which every pair of parents results in exactly two offspring, a local cut-and-splice operator probabilistically results in between 1 and 4 offspring for each pair of parents. Because of this uncertainty, it is convenient to recursively define the population produced by a cut-and-splice operator.

Definition 2.6.9 (Cut-and-splice operator): *Let I be an lfGA individual space, $\mu \in \mathbb{Z}^+$ (the* parent *population size), $\mu' \in \mathbb{Z}^+$ (the* offspring population size*), $\xi \triangleq \lceil \frac{2\mu'}{\mu} \rceil$, $\Omega \triangleq \pi_\mu^\xi \times ([0,1]^4 \times [0,1]^3)^{\mu'}$, $\omega \triangleq ((\sigma_1,\dots,\sigma_\xi),(\omega_1,\dots,\omega_{\mu'})) \sim U(\Omega)$, r' a local cut-and-splice operator, $r \in \mathcal{EVOP}(I,\mu,\mathbb{R}^2,\Omega)$, and*

$$\hat{r}(P'; i_0, i, j, k; P, p_c, p_s, \omega) \triangleq \begin{cases} P' & \text{, if } k = 0 \\ \hat{r}(P'; i_0, i_0, j-1, k; P, p_c, p_s, \omega) & \text{, if } k > 0 \text{ and } i = 0 \\ \hat{r}\left(\begin{array}{l} P' \cup \{P_{\sigma_j(i)}\}; \\ i_0, i_0, j-1, k-1; \\ P, p_c, p_s, \omega \end{array} \right) & \text{, if } k > 0 \text{ and } i = 1 \\ P' \cup \{Q_1, \dots, Q_k\} & \text{, if } 0 < k < 4 \text{ and } i > 1 \\ \hat{r}\left(\begin{array}{l} P' \cup \{Q_1, \dots, Q_{\dim \mathbf{Q}}\}; \\ i_0, i - \dim \mathbf{Q}, j, k - \dim \mathbf{Q}; \\ P, p_c, p_s, \omega \end{array} \right) & \text{, if } k > 0 \text{ and } i > 1 \end{cases} . \tag{3}$$

where $\mathbf{Q} = (Q_1, \dots, Q_{\dim \mathbf{Q}})$ denotes the offspring $[r'_{(p_c,p_s)}(\omega_i)](P_{\sigma_j(i)}, P_{\sigma_j(i-1)})$ of an invocation of r'.

If for every $p_c \in [0,1]$ *(the* cut probability*), every* $p_s \in [0,1]$ *(the* splice probability*), and every* $P \in I^\mu$ *(the* parent population*),* r *satisfies*

$$r_{(p_c,p_s)}(P) \quad = \quad \hat{r}(\{\}; \mu, \mu, \xi, \mu'; P, p_c, p_s, \omega) \quad ,$$

then r *is called a* cut-and-splice operator. *If* r' *is Goldberg's local cut-and-splice operator, then* r *is called* Goldberg's cut-and-splice operator. □

2.6.2.2 Mutation. The fast messy genetic algorithm uses a *building block filtering* (BBF) operator, which this research views as a mutation operator. The resulting population transformations map parent populations $P \in I(\lambda_0)$ to offspring populations $P' \in I(\lambda_f)$ where $\lambda_f \leq \lambda_0$. The mapping "deletes" $\lambda_0 - \lambda_f$ randomly chosen genes from each individual in P. The genes to be deleted from each individual are chosen uniformly without replacement. Equivalently, the genes to be retained are chosen uniformly without replacement. It is convenient to define the BBF operator in terms of the local BBF operator.

Definition 2.6.10 (Local building block filtering operator): *Let* I *be an lfGA individual space over genic alphabet* $\mathcal{A}$ *with nominal string length* ℓ,

$$\Omega \triangleq \left\{ \hat{\sigma} \in \mathcal{T}\left(\{0, \ldots, \ell\}, \bigcup_{i=1}^{\ell} \pi_i \right) : \hat{\sigma}(i) \in \pi_i \right\} \quad ,$$

$\omega \sim U(\Omega)$, *and* $m' \in \mathcal{EVOP}(I, 1, \{0, \ldots, \ell\}, \Omega)$ *an evolutionary operator. If for every* $\lambda_f \in \{0, \ldots, \ell\}$ *(the* offspring individual length*), and every* $\mathbf{a} = ((a_1, l_1), \ldots, (a_{\lambda_0}, l_{\lambda_0})) \in I$, m' *satisfies*

$$m'_{\lambda_f}(\mathbf{a}) = ((a_{[\omega(\lambda_f)](1)}, l_{[\omega(\lambda_f)](1)}), \ldots, (a_{[\omega(\lambda_f)](\lambda_f)}, l_{[\omega(\lambda_f)](\lambda_f)})) \quad ,$$

then m' *is called a* generalized local building block filtering operator. □

A local BBF operator is illustrated in Figure 10.

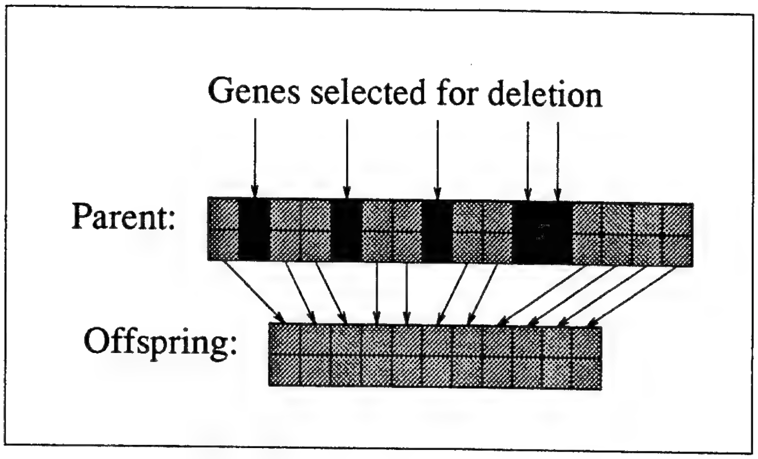


Figure 10. Local Building Block Filtering Operator

Definition 2.6.11 (Building block filtering operator): *Let I be an lfGA individual space with nominal string length ℓ, $\mu = \mu' \in \mathbb{Z}^+$ (the* population size*), $\lambda_0 \in \{1, \ldots, \ell\}$ (the* parent individual length*), $I(\lambda_0)$ defined by Equation 2, $\Omega \triangleq ([0,1]^{\lambda_0})^{\mu'}$, $\omega \triangleq (\omega_1, \ldots, \omega_{\mu'}) \sim U(\Omega)$, m' a local BBF operator, and $m : \mathbb{N} \longrightarrow \mathcal{T}(\Omega, \mathcal{T}(I^{\mu'}, (I(\lambda_f))^{\mu'}))$ an evolutionary operator. If for every $\lambda_f \in \{0, \ldots, \lambda_0\}$ (the* offspring individual length*), every $P \in (I(\lambda_0))^{\mu'}$ (the parent population), and every $i \in \{1, \ldots, \mu'\}$, m satisfies*

$$[m_{\lambda_f}(P)]_i = m'_{\lambda_f}(P_i) \quad ,$$

then m is called a building block filtering operator. □

Because the offspring individual lengths λ_f are deterministic (and identical for all individuals in the offspring population), this research sometimes refers to building block filtering operators as *deterministic building block filtering operators.* This is in contrast to *probabilistic building block filtering operators*, which are defined in Chapter III.

2.6.2.3 Selection. Both the messy genetic algorithm and the fast messy genetic algorithm use a selection operator called *tournament selection.* In its most general form, tournament selection can be described as follows:

1. Randomly draw q *competing individuals* from the current population.

2. Rank the competing individuals according to fitness.

3. Randomly draw one of the individuals (the *winner*) and include it in the next population.

By far the most frequently encountered form of tournament selection is binary tournament selection (BTS), for which $q \triangleq 2$.

Definition 2.6.12 (Binary tournament selection operator): *Let I be a non-empty set (the* individual space*), $\mu \in \mathbb{Z}^+$ (the* parent population size*), $\mu' \in \mathbb{Z}^+$ (the* offspring population size*), $\Omega \triangleq (\{1,\dots,\mu\}^2)^{\mu'}$, $\omega \triangleq ((\omega_0(1),\omega_1(1)),\dots,(\omega_0(\mu'),\omega_1(\mu')) \sim U(\Omega)$, and $s \in \mathcal{EVOP}(I,\mu,\mathcal{T}(I,\mathbb{R}),\Omega)$. If for every fitness function $\Phi : I \longrightarrow \mathbb{R}$ and every population $P \in I^\mu$, s satisfies*

$$[s_\Phi(P)]_i = \begin{cases} P_{\omega_0(i)} & , \textit{if } \Phi(P_{\omega_0(i)}) \geq \Phi(P_{\omega_1(i)}) \\ P_{\omega_1(i)} & , \textit{if } \Phi(P_{\omega_1(i)}) > \Phi(P_{\omega_0(i)}) \end{cases} ,$$

then s is called a binary tournament selection operator. □

Many variations of tournament selection are in common use. Some variations differ in the method by which the competing individuals are drawn from the population. If they are drawn without replacement, then they are typically drawn from a single "copy" of the population.

Other variations differ in the method by which the winner is drawn from the competing individuals. Typically, the winner is the most fit of the competing individuals (in which case implementation of the ranking step is unnecessary). Variations in which the winner is chosen according to some (non-trivial) probability density function defined on the rankings are called *probabilistic tournament selection* [33].

Finally, some variations use *thresholding*, which restricts the choice of competing individuals to those which are *compatible* with each other [36]. Individuals are considered compatible if they are sufficiently similar.

Definition 2.6.13 (Individual similarity, θ-compatible): *Let I be a non-empty set (the* individual space*). Then a mapping $d : I^2 \longrightarrow \mathbb{N}$ is called an* individual similarity*. Let $\mathbf{a}, \mathbf{b} \in I$ and $\theta : I^2 \longrightarrow \mathbb{N}$ (the* threshold mapping*). If $d(\mathbf{a}, \mathbf{b}) \geq \theta(\mathbf{a}, \mathbf{b})$ then $\mathbf{a}$ and $\mathbf{b}$ are θ-*compatible. □

For efficiency, implementations of BTS typically consider a maximum of $n_{sh} \in \mathbb{Z}^+$ (the *shuffle size*) individuals in seeking a compatible second individual.

Definition 2.6.14 (Binary tournament selection with thresholding operator and finite shuffle size n_{sh}): *Let I be a non-empty set (the* individual space*), $\ell \in \mathbb{Z}^+$, $\mu \in \mathbb{Z}^+$ (the* parent population size*), $\mu' \in \mathbb{Z}^+$ (the* offspring population size*), $n_{sh} \in \mathbb{Z}^+$ (the* shuffle size*), $\Omega \triangleq (\{1, \ldots, \mu\}^{n_{sh}})^{\mu'}$,*

$$\omega \triangleq ((\omega_0(1), \ldots, \omega_{n_{sh}}(1)), \ldots, (\omega_0(\mu'), \ldots, \omega_{n_{sh}}(\mu')) \sim U(\Omega) \quad ,$$

d an individual similarity, and $s \in \mathcal{EVOP}(I, \mu, \mathcal{T}(I^2, \mathbb{N}) \times \mathcal{T}(I, \mathbb{R}), \Omega)$. Also, define $j : \{1, \ldots, \mu'\} \times \mathcal{T}(I^2, \mathbb{N}) \longrightarrow \{0, \ldots, n_{sh}\}$ by

$$j(i, \theta) \triangleq \begin{cases} 0 & , \textit{if } (\forall k)[d(P_{\omega_0(i)}, P_{\omega_k(i)}) < \theta(P_{\omega_0(i)}, P_{\omega_k(i)})] \\ \min\{k : d(P_{\omega_0(i)}, P_{\omega_k(i)}) \geq \theta(P_{\omega_0(i)}, P_{\omega_k(i)})\} & , \textit{otherwise} \end{cases}$$

If for every $\theta : I^2 \longrightarrow \mathbb{N}$ (the threshold mapping*), every fitness function $\Phi : I \longrightarrow \mathbb{R}$, and every population $P \in I^\mu$, s satisfies*

$$[s_{(\theta, \Phi)}(P)]_i = \begin{cases} P_{\omega_0(i)} & , \textit{if } \Phi(P_{\omega_0(i)}) \geq \Phi(P_{\omega_{j(i,\theta)}(i)}) \\ P_{\omega_{j(i,\theta)}(i)} & , \textit{otherwise} \end{cases} ,$$

then s is called a binary tournament selection with thresholding operator. □

Because each is order-based, BTS and BTS with thresholding are examples of strictly invariant selection operators (see Section 2.5). Consequently, for evolutionary algorithms using either BTS or BTS with thresholding, effectiveness is unaffected by the choice of (order-preserving) fitness scaling function.

2.6.3 Messy Genetic Algorithms.

The dependence of the simple genetic algorithm's effectiveness on the "order" in which the genes are mapped to the object variables motivated Goldberg, et al. to propose the *messy genetic algorithm* (mGA) [31, 32, 36]. The mGA uses the order-invariant representation scheme defined in Section 2.6.1, as well as the strictly invariant BTS with thresholding operator. The algorithm is designed to obtain, with high probability, an order-k optimal individual (i.e. a fully-specified individual for which the fitness cannot be improved by changing k or fewer alleles), given an order-$(k-1)$ optimal competitive template. The parameter k is called the *building block size.*

Goldberg, et al. [31] suggest that the algorithm be applied iteratively for $1 \leq k \leq k_{\max}$, using the best individual found in iteration $k-1$ as the competitive template for iteration k. They also suggest [36] that $k_{\max}$ be "chosen to encompass the highest order deceptive nonlinearity suspected in the subject problem."[12] Such an estimate is typically not available. This author suggests that $k_{\max}$ must be viewed as controlling a tradeoff between expected solution quality (effectiveness) and execution time (efficiency).

The mGA consists of the *initialization*, *primordial*, and *juxtapositional* phases (see Figure 11). In the initialization phase, a deterministic technique called *Partially Enumerative Initialization* (PEI) produces an initial population containing at least one "copy" of each order-k potential building block. That is, for each order-k potential building block $\{(a_1, l_1), \ldots, (a_k, l_k)\}$, the initial population contains at least one individual of the form $\mathbf{a} \triangleq ((a_{\sigma(1)}, l_{\sigma(1)}), \ldots, (a_{\sigma(k)}, l_{\sigma(k)}))$, where σ is a permutation on $\{1, \ldots, k\}$. The usual initial

[12] The class of deceptive functions may be defined as follows (without loss of generality, a maximization problem is assumed). Let $I \triangleq \{0,1\}^\ell$, $\mathcal{L} \triangleq \{1, \ldots, \ell\}$, $\Phi \triangleq f \circ D : I \longrightarrow \mathbb{R}$ such that there exists a global maximum $f(\hat{\mathbf{x}})$ of f, D is one-to-one, and $(\exists \hat{\mathbf{a}} \in I)[D(\hat{\mathbf{a}}) = \hat{\mathbf{x}}]$. Also, for each *schema* (plural *schemata*) $\mathbf{h} = (h_1, \ldots, h_\ell) \in \{0, 1, \#\}^\ell$ define $o(\mathbf{h}) \triangleq \text{card}\,(\{i \in \mathcal{L} : h_i \neq \#\})$ and

$$\mathcal{S}(\mathbf{h}) \triangleq \{\mathbf{a} = (a_1, \ldots, a_\ell) \in I : (\forall i \in \mathcal{L})[h_i = \# \vee a_i = h_i]\} \ .$$

If $\mathcal{S}(\mathbf{h})$ contains $\hat{\mathbf{a}}$ and the individuals in $\mathcal{S}(\mathbf{h})$ have lower average fitness than the individuals in $\mathcal{S}(\mathbf{h}')$ for each of the "competing" schemata $\mathbf{h}'$, i.e. if

$$(\hat{\mathbf{a}} \in \mathcal{S}(\mathbf{h})) \wedge (\hat{\mathbf{a}} \notin \mathcal{S}(\mathbf{h}')) \wedge (h_i = \# \iff h'_i = \#) \Longrightarrow \left(\frac{\sum_{\mathbf{a} \in \mathcal{S}(\mathbf{h})} \Phi(\mathbf{a})}{\text{card}\,(\mathcal{S}(\mathbf{h}))} < \frac{\sum_{\mathbf{a} \in \mathcal{S}(\mathbf{h}')} \Phi(\mathbf{a})}{\text{card}\,(\mathcal{S}(\mathbf{h}'))} \right) \ ,$$

then f is called *deceptive* with respect to $\mathbf{h}$. The *order of deceptiveness* of f is $\max\{o(\mathbf{h}) : f \text{ is deceptive w.r.t. } \mathbf{h}\}$, and f is *order-k fully deceptive* if f is deceptive with respect to every schema $\mathbf{h}$ such that $\hat{\mathbf{a}} \in \mathcal{S}(\mathbf{h})$ and $o(\mathbf{h}) \leq k$.

Grefenstette [39] shows "that deception is neither necessary nor sufficient to make a problem difficult for GAs." This result in no way argues against the use of either order-invariant representations or strictly invariant selection operators.

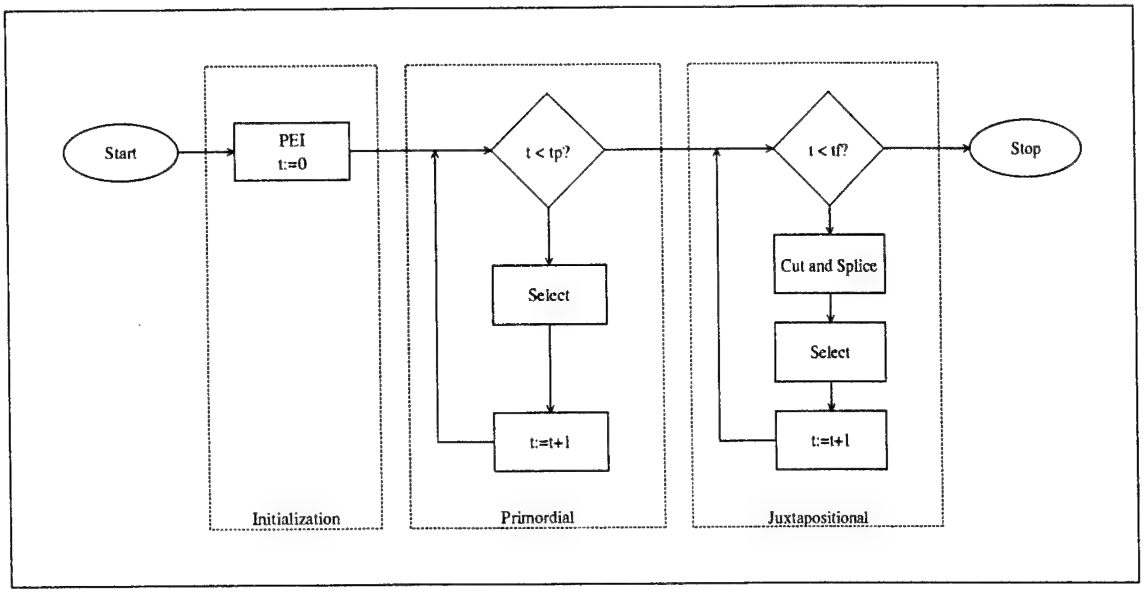


Figure 11. Messy Genetic Algorithm Flow Chart

population is

$$P(0) \triangleq \{((a_1, l_1), \ldots, (a_k, l_k)) \in I(k) : l_1 < \cdots < l_k\} \quad .$$

For a nominal string length ℓ, and a necessarily finite genic alphabet $\mathcal{A}$, the initial population contains

$$\mu^{(0)} \triangleq [\mathrm{card}\,(\mathcal{A})]^k \binom{\ell}{k}$$

individuals. Consequently. for the usual case of $\ell \gg k$, the algorithmic complexity of the initialization phase is $O([\ell \cdot \mathrm{card}\,(\mathcal{A})]^k)$. which is also the complexity of the overall algorithm. For $k \geq 3$, $\mu^{(0)}$ is much larger than typical simple genetic algorithm population sizes.

The primordial phase is designed to transform the initial population into a population of individuals $P(t_p) \subseteq I(k)$ which can be processed effectively and efficiently in the juxtapositional phase. The only operator used in the primordial phase is binary tournament selection with thresholding, with periodically decreasing offspring population sizes.[13] The individual similarity θ used by the mGA is such that, for each pair of

[13] Goldberg, et al. suggest that the competing individuals be drawn without replacement [36].

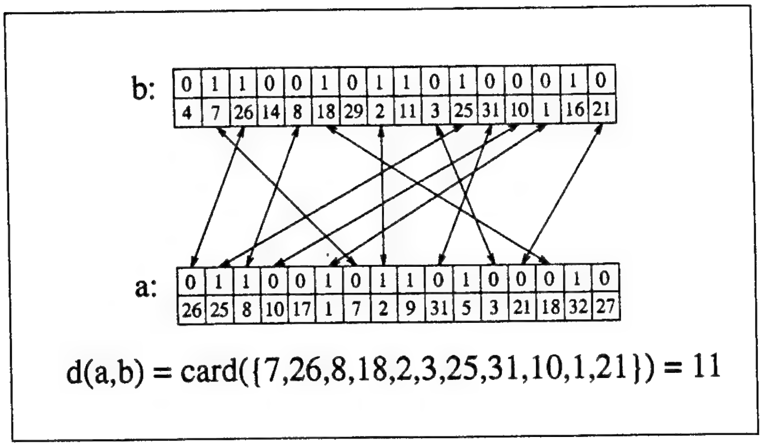


Figure 12. Two mGA individuals are depicted as vectors of allele-locus pairs. The individual similarity is the number of loci which occur at least once in each individual.

individuals $(\mathbf{a}, \mathbf{l}) = ((a_1, \ldots, a_{\lambda_a}), (l_1, \ldots, l_{\lambda_a})) \in \mathcal{A}^{\lambda_a} \times \mathcal{L}^{\lambda_a}$ and $(\mathbf{b}, \mathbf{m}) = ((b_1, \ldots, b_{\lambda_b}), (m_1, \ldots, m_{\lambda_b})) \in \mathcal{A}^{\lambda_b} \times \mathcal{L}^{\lambda_b}$

$$\theta((\mathbf{a}, \mathbf{l}), (\mathbf{b}, \mathbf{m})) \overset{\triangle}{=} \text{card}\left(\{l_1, \ldots, l_{\lambda_a}\} \cap \{m_1, \ldots, m_{\lambda_b}\}\right) \quad .$$

This is illustrated in Figure 12. The number of common defining loci of individuals $\mathbf{a} \sim U(I(\lambda_a))$ and $\mathbf{b} \sim U(I(\lambda_b))$ is a random variable X with the hypergeometric probability density function $h(\cdot; \lambda_a, \lambda_b; \ell)$. Individuals of length λ_a and λ_b are considered compatible if they share at least $\mathcal{E}[X] = \lambda_a \lambda_b / \ell$ common defining loci.

Finally, the juxtapositional phase uses Goldberg's cut-and-splice operator, as well as BTS with thresholding. Cut and splice probabilities are chosen to promote rapid increase of the individual length from k to ℓ [36]. The individual similarity and threshold mapping are the same as those used in the primordial phase.

Definition 2.6.15 (Messy genetic algorithm): *Let*

- *I be an lfGA individual space over the genic alphabet $\{0, 1\}$ with nominal string length ℓ and overflow factor o,*
- *$I(\lambda)$ defined by Equation 2,*
- *$k \in \{1, \ldots, \ell\}$ (the* building block size*),*

- $t_f \in \mathbb{Z}^+$ *(the* final generation*)*.

- $t_p \in \{0, \ldots, t_f\}$ *(the* final primordial phase generation*)*.

- $\mu^{(0)} \triangleq 2^k \binom{\ell}{k}$,

- $\{\mu^{(t)}\}_{t=0}^{t_f} \subset \mathbb{Z}^+$ *a non-increasing sequence (the* parent population sizes*)*,

- $\mu'^{(t)} \triangleq \mu^{(t+1)}$ *for* $t \in \{0, \ldots, t_f - 1\}$ *(the* offspring population sizes*)*.

- $\mathbf{c} \in I_F \triangleq I(\ell)$ *(the* competitive template),

- $\Phi_{\mathbf{c}} : I \longrightarrow \mathbb{R}$ *an lfGA fitness function*,

- $\iota : \bigcup_{i=1}^{\infty} (I^{\mu})^i \longrightarrow \{\texttt{true}, \texttt{false}\}$ *(the termination criterion) such that*

$$\iota(\{P(0), \ldots, P(t)\}) = \texttt{true} \iff \mathit{card}(\{P(0), \ldots, P(t)\}) > t_f \quad ,$$

- r *a sequence* $\{r^{(t)}\}$ *of Goldberg's cut-and-splice operators* $r^{(t)} : \mathbb{R}^2 \longrightarrow \mathcal{T}\left(\Omega_r^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu^{(t)}}\right)\right)$.

- m *a sequence* $\{m^{(t)}\}$ *of identity evolutionary operators.*

- s *a sequence* $\{s^{(t)}\}$ *of BTS with thresholding operators*

$$s^{(t)} : \mathcal{T}(I^2, \mathbb{N}) \times \mathcal{T}(I, \mathbb{R}) \longrightarrow \mathcal{T}\left(\Omega_s^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu'^{(t)}}\right)\right) \quad ,$$

- $p_c^{(t)} = p_s^{(t)} \triangleq 0$ *for* $0 \leq t < t_p$.

- $\Theta_r^{(t)} \triangleq (p_c^{(t)}, p_s^{(t)}) \in \mathbb{R}^2$ *for* $0 \leq t < t_f$ *(the* cut-and-splice parameters*). and*

- *the threshold mapping* $\theta : I^2 \longrightarrow \mathbb{N}$ *defined such that for* $\mathbf{a} \in (\mathcal{A} \times \mathcal{L})^{\lambda_a}$ *and* $\mathbf{b} \in (\mathcal{A} \times \mathcal{L})^{\lambda_b}$

$$\theta(\mathbf{a}, \mathbf{b}) \triangleq \left\lceil \frac{\lambda_a \lambda_b}{\ell} \right\rceil \quad .$$

Then the algorithm shown in Figure 13 is called a messy genetic algorithm. □

```
$t := 0$:
initialize $P(0) := \{\mathbf{a}_1(0), \ldots, \mathbf{a}_{\mu^{(0)}}(0)\} = \{\mathbf{a} = ((a_1, l_1), \ldots, (a_k, l_k)) \in I(k) : l_1 < \cdots < l_k\}$;
while ($\iota(\{P(0), \ldots, P(t)\}) \neq$ true) do
    recombine: $P'(t) := r_{\Theta_r^{(t)}}(P(t))$;
    mutate: $P''(t) := m(P'(t))$;
    select: $P(t+1) := s_{(\theta, \Phi_c)}(P''(t))$:
    $t := t + 1$;
od
```

Figure 13. Outline of a Messy Genetic Algorithm

2.6.4 Fast Messy Genetic Algorithms. The large initial population size of the messy genetic algorithm and the corresponding algorithmic complexity motivated Goldberg, et al. [35] to propose the fast messy genetic algorithm (fmGA), which is illustrated in Figure 14. The initial population of the fmGA is constructed using a technique called *Probabilistically Complete Initialization* (PCI), which randomly samples individuals from $I(\ell')$, where $\ell' \triangleq \ell - k$. The population size is chosen according to the population sizing relation of Goldberg, et al. [34], so that each order-k potential building block receives an expected number of "copies" sufficient to overcome sampling noise with specified probability.[14]

The goal of the fmGA primordial phase is the same as that of the mGA primordial phase. i.e. to obtain a population of individuals $P(t_p) \subseteq I(k)$, some of which can with high probability be juxtaposed to obtain an order-k optimal individual. Because the initial population consists of individuals $P(0) \subseteq I(\ell')$, building block filtering (BBF) is used to periodically reduce the lengths of the individuals (it is assumed that $\ell' \gg k$).

Definition 2.6.16 (Fast messy genetic algorithm): *Let*

- *I be an lfGA individual space over the genic alphabet $\{0,1\}$ with nominal string length ℓ and overflow factor o,*
- *$I(\lambda)$ defined by Equation 2,*
- *$k \in \{1, \ldots, \ell\}$ (the* building block size*),*

[14] In this context, a "copy" of the order-k potential building block $\{(a_1, l_1), \ldots, (a_k, l_k)\}$ is an individual of the form $((a_{\sigma(1)}, l_{\sigma(1)}), \ldots, (a_{\sigma(k)}, l_{\sigma(k)}))$, where σ is a permutation on $\{1, \ldots, \ell'\}$.

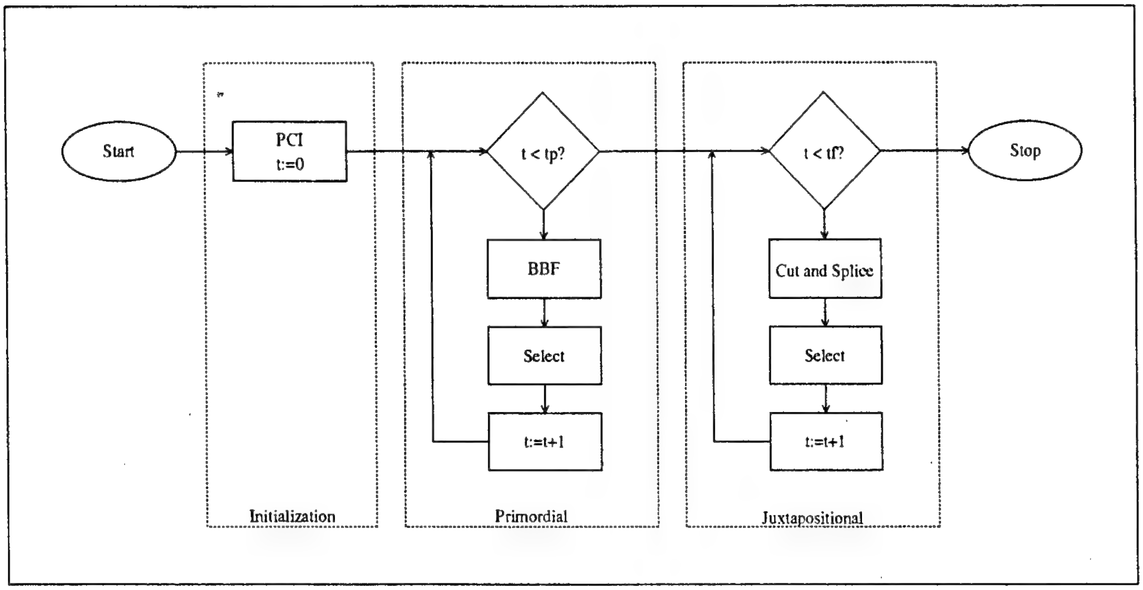


Figure 14. Fast Messy Genetic Algorithm Flow Chart

- $t_f \in \mathbb{Z}^+$ *(the* final generation*),*
- $t_p \in \{0, \ldots, t_f\}$ *(the* final primordial phase generation*).*
- $\lambda^{(0)} \triangleq \ell - k$ *(the* initial individual length*),*
- $\{\lambda^{(t)}\}_{t=0}^{t_p} \subset \{k, \ldots, \ell - k\}$ *a non-increasing sequence (the* individual lengths*),*
- $\alpha \in [0, 1]$ *(the* probability of selection error*),*
- $z_\alpha \in \mathbb{R}$ *such that* $Z \sim N(0, 1) \Longrightarrow \Pr[Z \geq z_\alpha] = 1 - \alpha$.
- $\beta^2 \in \mathbb{R}^+$ *(the maximum* inverse signal-to-noise ratio per sub-function *to be detected),*
- $\mu = \mu' \triangleq \frac{\ell!(\ell-2k)!}{(\ell-k)!^2} 2 z_\alpha^2 \beta^2 (\lceil \frac{\ell}{k} \rceil - 1) 2^k$ *(the* population size*),*
- $\mathbf{c} \in I_F \triangleq I(\ell)$ *(the* competitive template),
- $\Phi_\mathbf{c} : I \longrightarrow \mathbb{R}$ *an lfGA fitness function,*

- $\iota : \bigcup_{i=1}^{\infty} (I^{\mu})^i \longrightarrow \{\texttt{true}, \texttt{false}\}$ *(the termination criterion) such that*

$$\iota(\{P(0), \ldots, P(t)\}) = \texttt{true} \Longleftrightarrow \mathit{card}\,(\{P(0), \ldots, P(t)\}) > t_f \quad ,$$

- r *a sequence* $\{r^{(t)}\}$ *of Goldberg's cut-and-splice operators* $r^{(t)} : \mathbb{R}^2 \longrightarrow \mathcal{T}\left(\Omega_r^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu^{(t)}}\right)\right)$.
- m *a sequence* $\{m^{(t)}\}$ *of evolutionary operators,*
- *for* $0 \leq t < t_p$, $m^{(t)} : \mathbb{N} \longrightarrow \mathcal{T}\left(\Omega_m^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu^{(t)}}\right)\right)$ *a BBF operator,*
- *for* $t_p \leq t < t_f$, $m^{(t)}$ *an identity evolutionary operator,*
- s *a sequence* $\{s^{(t)}\}$ *of BTS with thresholding operators,*

$$s^{(t)} : \mathcal{T}(I^2, \mathbb{N}) \times \mathcal{T}(I, \mathbb{R}) \longrightarrow \mathcal{T}\left(\Omega_s^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu'^{(t)}}\right)\right) \quad ,$$

- $\Theta_m^{(t)} \triangleq \lambda^{(t)}$ *for* $0 \leq t < t_p$ *(the* filtering parameters*),*
- $p_c^{(t)} = p_s^{(t)} \triangleq 0$ *for* $0 \leq t < t_p$,
- $\Theta_r^{(t)} \triangleq (p_c^{(t)}, p_s^{(t)}) \in \mathbb{R}^2$ *for* $0 \leq t < t_f$ *(the* cut-and-splice parameters*), and*
- θ *a sequence* $\{\theta^{(t)}\}$ *of threshold mappings* $\theta^{(t)} : I^2 \longrightarrow \mathbb{N}$.

Then the algorithm shown in Figure 15 is called a fast messy genetic algorithm. □

```
$t := 0$:
initialize $P(0) := \{\mathbf{a}_1(0), \ldots, \mathbf{a}_{\mu^{(0)}}(0)\} \sim U(I(\lambda^{(0)}))$;
while $(\iota(\{P(0), \ldots, P(t)\}) \neq \texttt{true})$ do
        recombine: $P'(t) := r_{\Theta_r^{(t)}}(P(t))$;
        mutate: $P''(t) := m_{\Theta_m^{(t)}}(P'(t))$;
        select: $P(t+1) := s_{(\theta, \Phi_c)}(P''(t))$;
        $t := t + 1$:
od
```

Figure 15. Outline of a Fast Messy Genetic Algorithm

The hope is that a balance can be found between the the explorative effect of filtering and exploitative effect of selection. This balance depends on the specification of the sequences of individual lengths $\{\lambda^{(t)}\}$ and threshold mappings $\{\theta^{(t)}\}$. Goldberg, et al. suggest the more conservative threshold mappings $\theta^{(t)} : I^2 \longrightarrow \mathbb{N}$ such that for $(\mathbf{a}, \mathbf{b}) \in (\mathcal{A} \times \mathcal{L})^{\lambda_a} \times (\mathcal{A} \times \mathcal{L})^{\lambda_b}$, and independent of t,

$$\theta(\mathbf{a}, \mathbf{b}) \triangleq \left\lceil \frac{\lambda_a \lambda_b}{\ell} + 3\sqrt{\frac{\lambda_a(\ell - \lambda_a)\lambda_b(\ell - \lambda_b)}{\ell^2(\ell - 1)}} \right\rceil .$$

This and other existing fmGA parameter selection techniques are discussed in Section 2.6.5. None reliably obtains the necessary balance between convergence and disruption in practical applications.

2.6.5 fmGA Parameter Selection Techniques. The effectiveness of the fast messy genetic algorithm for a given application depends on a number of design parameters. In particular, experience [28] shows that the effectiveness of the algorithm is highly sensitive to the sequences of individual lengths $\{\lambda^{(t)}\}$ and threshold mappings $\{\theta^{(t)}\}$. No previously proposed techniques for selection of these parameters [28, 35, 46, 47, 54] reliably yields satisfactory effectiveness in practical applications.

Each of these techniques is based on the premise that in order to be effective, the algorithm must produce a final primordial phase population which contains "building blocks" in proportions sufficient to ensure "good mixing" in the juxtapositional phase. In this context (and thus in the remainder of this section), *building blocks* are those order-k potential building blocks with juxtapose to form "the" order-k optimal individual. Where no unique order-k optimal individual exists, "building blocks" are not well defined.

The earliest techniques [27, 28, 35, 46, 54] are essentially heuristic (Section 2.6.5.1). Kargupta's more recent methodology [47] is less heuristic and yields parameters resulting in improved effectiveness in a limited study. It is based on a more complete model of tournament selection than the earlier techniques (Section 2.6.5.2). None of the techniques predicts the expected effectiveness of the algorithm, nor whether improved effectiveness may result from "tweaking" the parameters.

2.6.5.1 Heuristic Techniques. The technique proposed by Goldberg, et al. in their fmGA study [35] is based on the heuristic that an adequate final primordial phase population results when each selection episode produces a sufficient number of copies of each building block to prevent extinction by BBF. Based on the probability of building block survival, a *building-block repetition factor* of

$$\begin{aligned} \gamma &\triangleq \frac{\binom{\lambda^{(i-1)}}{\lambda^{(i)}}}{\binom{\lambda^{(i-1)}-k}{\lambda^{(i)}-k}} \\ &\approx \left(\frac{\gamma^{(i-1)}}{\gamma^{(i)}}\right)^{k}, \quad \text{for } \gamma^{(i)} \gg k \\ &= \rho_i^{-k} , \end{aligned}$$

where $\rho_i \triangleq \frac{\gamma^{(i)}}{\gamma^{(i-1)}}$, is sufficient for at least one copy of an order-k building block to survive a reduction of string length from $\lambda^{(i-1)}$ to $\lambda^{(i)}$.

The fmGA study proposes "fixing γ to a constant value much less than 2^{t_s}, where t_s is the number of selection repetitions per length reduction." Doing so "roughly implies a fixed length-reduction ration $\rho = \rho_i$ for all i." It is not clear how t_s should be chosen, nor is it clear how to choose ρ except that $\gamma \ll 2^{t_s}$ should be satisfied.

Regarding the use of the thresholding parameters proposed earlier by Goldberg, Deb, and Korb [31], the fmGA study reports that "this procedure has not proven to be adequate." Instead, a threshold of

$$\theta = \left\lceil \frac{\lambda_1 \lambda_2}{\ell} + 3\sigma \right\rceil$$

is suggested, where λ_1 and λ_2 are the lengths of the competing individuals, ℓ is the nominal string length, and σ^2 is the variance of the hypergeometric distribution [53] having parameters λ_1, λ_2, and ℓ:

$$\sigma^2 = \frac{\lambda_1(\ell - \lambda_1)\lambda_2(\ell - \lambda_2)}{\ell^2(\ell - 1)} .$$

The experiments reported in the fmGA study do not use parameters obtained using the proposed methodology. According to Kargupta [46],

> The [experimental] results presented in the ICGA paper were based on [an] empirically tuned schedule, since we did not have the complete theoretical analysis of [the fast messy genetic algorithm] at that time.

The empirical tuning involves measuring the fraction of the individuals in each generation containing each building block, and adjusting the parameters based on those fractions [46]. Parameters obtained via this method for a 50-bit order-5 fully deceptive function[15] are shown in Table 2. The specific tuning strategy by which the final parameters are obtained is not known to this author.

Table 2. Empirically tuned fmGA thresholding and filtering parameters for a 50-bit order-5 fully deceptive objective function

Episode	Cut generation	String length	Threshold
0	0	45	39
1	7	39	35
2	11	34	28
3	15	29	23
4	19	25	18
5	23	22	15
6	29	19	13
7	35	16	10
8	41	14	9
9	47	12	7
10	53	10	6
11	59	8	5
12	65	7	4
13	71	6	3
14	77	5	4

This empirical tuning method requires *a priori* knowledge of which genes constitute building blocks. Such knowledge is not available for practical applications. Consequently, this parameter selection technique is not generally applicable. For example, in an application of the fast messy genetic algorithm to *energy minimization* [54], it is not known whether or not order-k building blocks exist, much less which genes constitute those building blocks. Furthermore, the execution time required for this application prohibits

[15] Although apparently not the same function addressed in the fmGA study.

any substantial empirical tuning of the exogenous parameters. The experiments reported in the energy minimization study use a heuristically "scaled version" of the schedule in Table 2, which is shown in Table 3. In another application to the same problem [28], the average effectiveness of the algorithm resulting from this schedule is compared to three others:

Table 3. Scaled fmGA thresholding and filtering parameters for a 240-bit objective function

Episode	Generation	String length	Threshold
0	0	216	194
1	7	185	143
2	11	157	107
3	15	135	84
4	19	115	64
5	23	98	47
6	29	84	38
7	35	72	31
8	41	61	25
9	47	53	21
10	53	45	17
11	59	39	15
12	65	33	12
13	71	29	10
14	77	24	8
15	82	21	7
16	87	18	6
17	92	15	5
18	97	13	4
19	102	11	4
20	107	9	3
21	112	8	3
22	117	7	2
23	122	6	2
24	127	5	3

1. "50% initial similarity, linearly increasing to 100% similarity," i.e. the threshold mapping in episode e is such that $\theta_{(e)} = (\frac{1}{2} + \frac{1}{2}\frac{e}{e_f})\lambda_{(e)}$ where e_f is the final selection episode;

2. "50% initial similarity, linearly increasing to 80% similarity," i.e. $\theta_{(e)} = (\frac{1}{2} + \frac{3}{10}\frac{e}{e_f})\lambda_{(e)}$; and

3. "constant 80% similarity," i.e. $\theta_{(e)} = \frac{4}{5}\lambda_{(e)}$.

The results obtained for this application, while far from exhaustive, indicate that the last schedule is the most effective of those compared.

2.6.5.2 Kargupta's Technique. Kargupta's more recent methodology [47] is less heuristic in nature and yields parameters resulting in improved effectiveness in a limited study compared to the method of Goldberg, et al. [29]. This section formalizes Kargupta's description of this technique. The stated design objective of the technique is to ensure that the fraction of individuals containing each of the building blocks grows nearly uniformly. This growth is achieved and controlled through the choice of three sets of design variables:

1. the duration $t^*_{(e)}$ of each selection episode e,
2. the threshold parameter $\theta_{(e)}$ for each selection episode e, and
3. the number of genes $\lambda_{(e)} - \lambda_{(e+1)}$ deleted in each filtering event.

Formal statement of the technique is facilitated by a brief review of the underlying theory.

The theoretical development includes a more "realistic" model of BTS with thresholding than that used in the research of Goldberg, et al. [35], focusing on "cross-competition" between building blocks. The model views individuals as containing no more than one of m building blocks. That is, individuals are viewed as belonging to one of $m + 1$ classes, where m is the number of building blocks:

- the classes "i", where $i \in \{1, \ldots, m\}$, consisting of the individuals containing building block i (assumed to be mutually exclusive), and
- the class "*junk*" of individuals containing no building blocks.

The fraction of individuals in class i in generation $t + 1$ is modeled by

$$q_{i,t+1} = q_{i,t} \left(2 - q_{i,t} - \sum_{j \neq i} (2 - \alpha_{ij}) q_{j,t} \right) .$$

and the fraction of individuals in *junk* is

$$\begin{aligned} q_{junk,t+1} &= q_{junk,t}^2 \\ &= \left(q_{junk,0}^2\right)^t \quad . \end{aligned}$$

where α_{ij} is "the expected number of copies of building block i resulting from competition with building block j." The matrix $\underline{\alpha}$ having components α_{ij} is called the *interaction matrix.*

Using this model, Kargupta considers two "extreme" cases. The first is the "unbiased" case, in which $\alpha_{ij} = 1$ for all i, j, corresponding to equally scaled building blocks, so that the interaction matrix is

$$\underline{\alpha} = \underline{\mathbf{1}} \triangleq \begin{bmatrix} 1 & 1 & \ldots & 1 \\ 1 & 1 & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 \end{bmatrix} \quad . \tag{4}$$

The second is the "strong bias" case, in which the interaction matrix is of the form

$$\underline{\alpha} \quad = \quad \left[\begin{array}{c|c|c} 1 & 2 & \mathbf{2}^\top \\ \hline 0 & 1 & \mathbf{0}^\top \\ \hline \mathbf{0} & \mathbf{2} & \underline{\mathbf{1}} \end{array}\right] \quad . \tag{5}$$

In the strong bias case, every individual which contains a particular building block i is more fit than every individual which lacks building block i (Equation 5 assumes without loss of generality that $i = 1$). Also, for some building block j, every individual which contains a building block $k \neq j$ is more fit than every individual which contains building block j (Equation 5 assumes, again without loss of generality, that $j = 2$). Excluding building blocks i and j, the strong bias case is identical to the unbiased case.

The theoretical development motivating Kargupta's parameter selection technique also includes a model of BBF, essentially identical to that of Goldberg, et al. [35]. The fraction of individuals in class i following the BBF event following selection episode e are modeled by

$$q_{i,0,e+1} = \eta_{e+1} q_{i,t_e^*,e} \quad ,$$

and the fraction of individuals in $junk$ is

$$q_{junk,0,e+1} = q_{junk,t_e^*,e} + (1-\eta_{e+1}) \sum_i q_{i,t_e^*,e} \quad ,$$

where

$$\eta_{e+1} = \frac{\binom{\lambda_{(e)}-k}{\lambda_{(e+1)}-k}}{\binom{\lambda_{(e)}}{\lambda_{(e+1)}}} \quad .$$

Based on these models of BTS and building block filtering, Kargupta proposes a methodology by which to obtain fast messy genetic algorithm exogenous parameters. The stated design principle motivating the technique is the control of "niche sizes." In brief, Kargupta seeks to choose the thresholds so that filtered individuals are $\theta_{(e+1)}$-compatible if and only if their parents are $\theta_{(e)}$-compatible, i.e.

$$\Lambda_c(\mathbf{a},\mathbf{b}) \geq \theta_{(e)} \Longleftrightarrow \Lambda_c(m(\mathbf{a}), m(\mathbf{b})) \geq \theta_{(e+1)}$$

is satisfied. The technique is apparently designed to satisfy the condition in expectation in some sense. It may be stated formally as shown in Figure 16. Kargupta views t_s. γ. δ. and β as design parameters, but offers little guidance as to how they should be selected for a given application. The experiments he reports are for an order-5 fully deceptive fitness function. with $\rho = 0.5$. $\delta = 0.01$ and $\beta = 2$, which implies that $\gamma = \rho^{-k} = 32$. Because $\gamma \ll 2^{t_s}$, Kargupta's experiments are apparently for $t_s > 5$.

1. Fix $\gamma \ll 2^{t_e}$, $\rho \triangleq \gamma^{-\frac{1}{k}}$, $\delta \in (0,1)$, and $\beta > 1$.
2. Take $\lambda_{(0)} \triangleq \ell - k$ and $\theta_{(0)} \triangleq \left\lceil \frac{(\lambda_{(0)})^2}{\ell} \right\rceil$.
3. For each episode e, with $\mathbf{a}, \mathbf{b} \sim U(I(\lambda_{(e)}))$:
    (a) Take $\lambda_{(e+1)} \triangleq \rho\lambda_{(e)}$.
    (b) Take $t_e^* \triangleq \max\{t \in \mathbb{Z}^+ : (\forall i,j)[|q_{i,t} - q_{j,t}| \leq \delta]\}$.
    (c) If $\lambda_{(e)} > \beta k$, take $\theta_{(e+1)} \triangleq \mathcal{E}\left[\Lambda_c(m(\mathbf{a}), m(\mathbf{b})) \mid \Lambda_c(\mathbf{a}, \mathbf{b}) < \theta_{(e)}\right]$
    (d) If $\lambda_{(e)} \leq \beta k$, take $\theta_{(e+1)} \triangleq \mathcal{E}\left[\Lambda_c(m(\mathbf{a}), m(\mathbf{b}))\right]$.

Figure 16. Kargupta's fmGA Parameter Selection Technique

The choice of the initial individual length $\lambda_{(0)}$ is consistent with the recommendations of Goldberg, et al. [35]. The individual lengths $\lambda_{(e)}$ resulting from the constant string reduction ratio ρ are also consistent with those recommendations. Similarly, the choice of the initial threshold $\theta_{(0)}$ is consistent with the original messy genetic algorithm thresholding theory [16].

Each t_e^* is chosen so that as many iterations of selection as possible are performed while ensuring "even growth" of the building blocks within episode e. Kargupta does not address the existence or determination of a $\delta \in (0,1)$ such that each $t_e^* \geq 1$ for a given application. If such a δ does not exist or simply cannot be readily identified, the technique fails.

The last two steps choose the threshold parameters heuristically so as to control the "niche sizes," as discussed previously. Strictly speaking, Kargupta's description of the technique specifies a choice of the threshold reduction $\lambda_{(e)} - \lambda_{(e+1)}$, rather than the threshold $\lambda_{(e+1)}$ itself. In the early episodes (for which θ is determined by the conditional expectation), the thresholds are chosen to be relatively small, thus permitting relatively unrestricted competition. In the late episodes (for which θ is determined by the unconditional expectation), the thresholds are more conservative, thus reducing cross competition.

Kargupta offers little justification for the specific expectations recommended, except that they resulted in improved effectiveness over previous scheduling techniques for the experiments performed. Nor does he address the fact that the conditional expectation fails to exist when $\theta_{(e)} \leq 2\lambda_{(e)} - \ell$, i.e. when all individuals are $\theta_{(e)}$-compatible.[16]

2.7 Summary

Evolutionary algorithms are a class of stochastic population-based algorithms which are commonly applied as optimum seeking techniques. Included within this broad class are the loosely defined classes of genetic algorithms, evolutionary programming. and evolution strategies. A novel framework for evolutionary algorithms is proposed. in which evolutionary operators are viewed as mappings from parameter spaces to random population transformations. Definitions of recombination, mutation. and selection operators which capture their distinguishing characteristics are proposed within this framework.

A specific example of evolutionary algorithms is the simple genetic algorithm (sGA). Another class of evolutionary algorithms, which historically arose from genetic algorithms research is the class of linkage-friendly genetic algorithms (lfGAs). The primary distinctions of lfGAs. as defined in this research, are their use of order-invariant representation schemes and strictly invariant selection operators (see Figure 17). Specific examples of lfGAs include the messy genetic algorithm (mGA). the fast messy genetic algorithm (fmGA). and the gene expression messy genetic algorithm (gemGA). The effectiveness of the fmGA is sensitive to the sequences of individual lengths and threshold mappings. Existing fmGA parameter selection techniques do not reliably yield satisfactory effectiveness for practical applications.

[16] Kargupta's actual recommendation for early episode thresholds is $\theta_{(e+1)} = \mathcal{E}\left[\Lambda_c(m(\mathbf{a}), m(\mathbf{b})) \mid \theta_{(e)} - (\lambda_{(e)} - \theta_{(e)}) \leq \Lambda_c(\mathbf{a}.\mathbf{b}) < \theta_{(e)}\right]$ but this seems particularly arbitrary and is inconsistent with other parts of the discussion. Furthermore, this fails to exist if $\theta_{(e)} \geq \lambda_{(e)}$. This condition occurs, for example, in the initial selection episode when $\lambda_{(0)} \gg k$ so that $\theta_{(0)} = \lambda_{(0)}$.

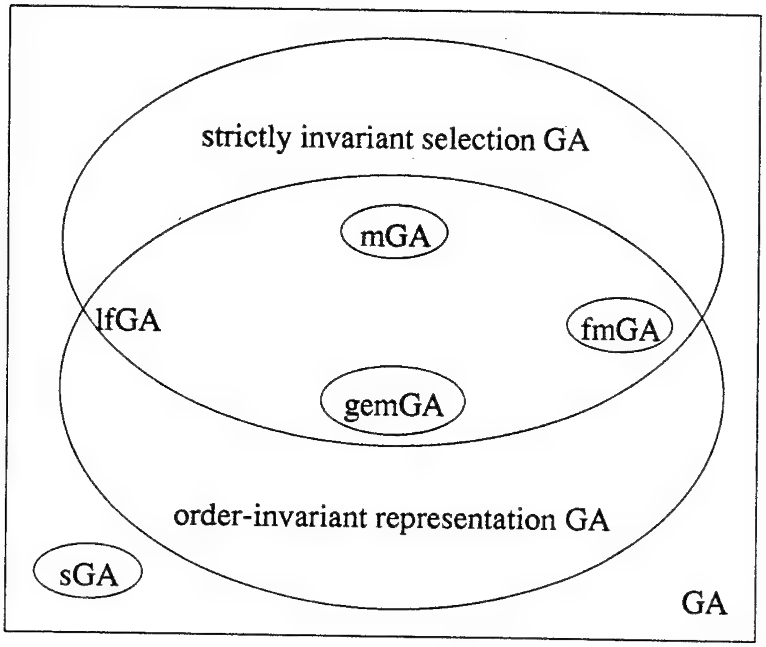

strictly invariant selection GA
mGA
lfGA
fmGA
gemGA
order-invariant representation GA
sGA
GA

Figure 17. Venn Diagram of the Class of Genetic Algorithms

III. Generalized Fast Messy Genetic Algorithms

This chapter proposes a novel linkage-friendly genetic algorithm. The algorithm shares the high-level structure of the fast messy genetic algorithm, shown in Figure 14, as well as the representation scheme of the messy genetic algorithm (mGA) and fast messy genetic algorithm (fmGA), defined in Section 2.6.1. Consequently, it is convenient to refer to the algorithm as the *generalized fast messy genetic algorithm* (gfmGA).

In the gfmGA initialization phase, a competitive template is selected and an initial population is randomly generated. The gfmGA primordial phase uses the probabilistic building block filtering operator (Section 3.1) and binary tournament selection with probabilistic thresholding operator (Section 3.2). Both of these operators are novel generalizations of the operators used by the fmGA. The juxtapositional phase uses the cut-and-splice operator (Section 2.6.2.1), as well as the binary tournament selection with probabilistic thresholding operator. Section 3.3 defines the gfmGA in the formal framework of Section 2.3 and shows that the fmGA is a special case of the gfmGA.

Mathematical models of the two gfmGA primordial phase operators are developed in Chapters IV and V. Together, the models permit the definition of expected gfmGA effectiveness as a continuously differentiable function of the gfmGA parameters (Chapter VI). Optimization of the related cost function J by various techniques yields parameter selection methodologies for the fmGA and the gfmGA.

Because the fmGA is a special case of the gfmGA, existence is guaranteed of parameters for which the gfmGA expected effectiveness is no worse than the best possible fmGA expected effectiveness.

Furthermore, partly because the gfmGA parameters are real-valued, vector space optimization techniques may be used to obtain formal necessary optimality conditions (NOCs) for the gfmGA parameters.

3.1 Mutation

The generalized fast messy genetic algorithm uses the *probabilistic building block filtering* (probabilistic BBF) operator, which this research views as a mutation operator. The probabilistic BBF operator is a novel generalization of the deterministic BBF operator (Section 2.6.2.2). Whereas the deterministic building block filtering operator deletes the same fixed number $\lambda^{(t)} - \lambda^{(t+1)}$ of genes from each individual in generation t, the probabilistic BBF operator adds or deletes a random number of genes. determined independently for each individual.

If $\lambda^{(t+1)} < \lambda^{(t)}$, the genes to be deleted are drawn without replacement from a uniform distribution over all of the individual's genes. Equivalently, the genes to be retained are chosen uniformly without replacement. If $\lambda^{(t+1)} > \lambda^{(t)}$, the genes to be added are generated by drawing without replacement from the set of loci for which the individual does not already contain a gene. then drawing from the genic alphabet independently for each new gene. Thus. the operator preserves the non-overspecified property of primordial phase individuals. It is convenient to define the probabilistic BBF operator in terms of the generalized local BBF operator.

Definition 3.1.1 (Generalized local building block filtering operator): *Let I be an lfGA individual space over genic alphabet $\mathcal{A}$ with nominal string length ℓ,*

$$\mathcal{S}_\pi \triangleq \left\{ \hat{\sigma} \in \mathcal{T}\left(\{0, \ldots, \ell\}, \bigcup_{i=1}^{\ell} \pi_i \right) : \hat{\sigma}(i) \in \pi_i \right\} \quad ,$$

$\Omega \triangleq \mathcal{S}_\pi \times \mathcal{S}_\pi \times \mathcal{A}^\ell$. $\omega \triangleq (\hat{\sigma}_1, \hat{\sigma}_2, (\alpha_1, \ldots, \alpha_\ell)) \sim U(\Omega)$. $sort(\{\beta_1, \ldots, \beta_\lambda\}) \triangleq (\beta_{n_1}, \ldots, \beta_{n_\lambda})$ *such that* $\beta_{n_1} < \ldots < \beta_{n_\lambda}$.

$$\begin{aligned} &\hat{m}[((a_1, l_1), \ldots, (a_{\lambda_0}, l_{\lambda_0})), (\alpha_1, \ldots, \alpha_\ell), (\beta_{\lambda_0+1}, \ldots, \beta_\ell), \sigma_1, \sigma_2, \lambda_f] \\ &\triangleq \begin{cases} ((a_{\sigma_1(1)}, l_{\sigma_1(1)}), \ldots, (a_{\sigma_1(\lambda_f)}, l_{\sigma_1(\lambda_f)})) & , \textit{if } \lambda_f \leq \lambda_0 \\ \Big((a_{\sigma_1(1)}, l_{\sigma_1(1)}), \ldots, (a_{\sigma_1(\lambda_f)}, l_{\sigma_1(\lambda_f)}), & \\ (\alpha_{\sigma_2(1)+\lambda_0}, \beta_{\sigma_2(1)+\lambda_0}), \ldots, (\alpha_{\sigma_2(\lambda_f-\lambda_0)}, \beta_{\sigma_2(\lambda_f-\lambda_0)}) \Big) & , \textit{if } \lambda_f > \lambda_0 \end{cases} \end{aligned} \quad ,$$

and $m' \in \mathcal{EVOP}(I, 1, \mathrm{N}, \Omega)$ *an evolutionary operator. If for every* $\lambda_f \in \{0, \dots, \ell\}$ *(the* offspring individual length*), and every* $\mathbf{a} = ((a_1, l_1), \dots, (a_{\lambda_0}, l_{\lambda_0})) \in I$*,* m' *satisfies*

$$m'_{\lambda_f}(\mathbf{a}) = \hat{m}(\mathbf{a}, (\alpha_1, \dots, \alpha_\ell), sort(\{1, \dots, \ell\} - \{l_1, \dots, l_{\lambda_0}\}), \hat{\sigma}_1(\lambda_0), \hat{\sigma}_2(\ell - \lambda_0), \lambda_f) \quad ,$$

then m' *is called a* generalized local building block filtering operator. □

The number of genes added or deleted from a parent individual in generation $t \in \{0, \dots, t_p - 1\}$ is determined by the offspring individual length $\lambda^{(t+1)}$, which is a random variable chosen according to

$$\psi^{(t)}(\lambda) \triangleq \Pr[\lambda^{(t+1)} = \lambda]$$

where each $\psi^{(t)}(\lambda)$ is an exogenous *filtering parameter.* Because each $\psi^{(t)}$ is a probability density function of the discrete type,

$$\psi^{(t)}(0) \triangleq 1 - \sum_{\lambda=1}^{\ell} \psi^{(t)}(\lambda) \quad ,$$

and the $\psi^{(t)}(\lambda)$'s are subject to the constraints

$$(\forall \lambda \in \{1, \dots, \ell\})[\psi^{(t)}(\lambda) \geq 0]$$

and

$$\sum_{\lambda=1}^{\ell} \psi^{(t)}(\lambda) \leq 1 \quad . \tag{6}$$

Definition 3.1.2 (Probabilistic building block filtering operator): *Let* I *be an lfGA individual space over genic alphabet* $\mathcal{A}$ *with nominal string length* ℓ*,* $\mu = \mu' \in \mathbb{Z}^+$ *(the* population size*),*

$$\mathcal{S}_\pi \triangleq \left\{ \hat{\sigma} \in \mathcal{T}\left(\{0, \dots, \ell\}, \bigcup_{i=1}^{\ell} \pi_i \right) : \hat{\sigma}(i) \in \pi_i \right\} \quad .$$

$\Omega \triangleq (\mathcal{S}_\pi \times \mathcal{S}_\pi \times \mathcal{A}^\ell)^{\mu'}$, $\omega \triangleq (\omega_1, \ldots, \omega_{\mu'}) \sim U(\Omega)$, *$m'$ a generalized local BBF operator, and* $m : [0,1]^\ell \longrightarrow \mathcal{T}(\Omega, \mathcal{T}(I^{\mu'}, (I(\lambda_f))^{\mu'}))$ *an evolutionary operator. If for every* $\psi \triangleq (\psi(1), \ldots, \psi(\ell))$ *satisfying Equations 6 (the* filtering parameters*), every* $P \in I^\mu$ *(the parent population), and every* $i \in \{1, \ldots, \mu'\}$, *$m$ satisfies*

$$\Pr\left\{[m_\psi(P)]_i = [m'_{\lambda_f}(\omega_i)](P_i)\right\} = \begin{cases} \psi(\lambda_f) & , \text{if } 1 \leq \lambda_f \leq \ell \\ 1 - \sum_{\lambda=1}^{\ell} \psi(\lambda) & , \text{if } \lambda_f = 0 \end{cases} ,$$

then m is called a probabilistic building block filtering operator. □

3.2 Selection

The gfmGA primordial phase uses the binary tournament selection (BTS) with probabilistic thresholding operator. As in the deterministic case (Section 2.6.2.3) competition is restricted to those which are determined to be *compatible*. In contrast to the deterministic case, individuals are considered compatible with a probability which depends on their similarity. The formal definition of probabilistic compatibility is more general in that it does not require the threshold mapping to depend on an individual similarity.

Definition 3.2.1 (Probabilistically θ-compatible): *Let I be a non-empty set (the* individual space*) and* $\theta : I^2 \longrightarrow [0,1]$ *(the* probabilistic threshold mapping*). Then* $\mathbf{a} \in I$ *and* $\mathbf{b} \in I$ *are called probabilistically θ-compatible with probability* $\theta(\mathbf{a}, \mathbf{b})$. *If* $X \sim U([0,1])$ *is sampled and* $X \leq \theta(\mathbf{a}, \mathbf{b})$ *then* **a** *and* **b** *are* found to be probabilistically θ-compatible. □

For the gfmGA, individuals $\mathbf{a} \in I(\lambda_a)$ and $\mathbf{b} \in I(\lambda_b)$ and sharing $\lambda_c = \Lambda_c(\mathbf{a}, \mathbf{b})$ common defining loci are probabilistically θ-compatible with probability $\theta^{(t)}(\lambda_c; \lambda_a, \lambda_b)$, where each $\theta^{(t)}(\lambda_c; \lambda_a, \lambda_b)$ is an exogenous *thresholding parameter.*

The thresholding parameters are probabilities for each $t \in \{0. \dots . t_f - 1\}$. and hence subject to the constraints

$$\begin{aligned} \forall\lambda, \lambda_a, \lambda_b &: \theta^{(t)}(\lambda; \lambda_a, \lambda_b) \geq 0 \quad , \\ \forall\lambda. \lambda_a, \lambda_b &: \theta^{(t)}(\lambda; \lambda_a . \lambda_) \leq 1 \quad . \end{aligned} \tag{7}$$

Definition 3.2.2 (Binary tournament selection with probabilistic thresholding operator and finite shuffle size n_{sh}): *Let I be a non-empty set (the* individual space*), $\ell \in \mathbb{Z}^+$, $\mu \in \mathbb{Z}^+$ (the* parent population size*), $\mu' \in \mathbb{Z}^+$ (the* offspring population size*), $n_{sh} \in \mathbb{Z}^+$ (the* shuffle size*), $\Omega \triangleq (\{1, \dots, \mu\}^{n_{sh}})^{\mu'} \times [0, 1]^{n_{sh} \times \mu'}$,*

$$\omega \triangleq ((\omega_0(1), \dots, \omega_{n_{sh}}(1)), \dots, (\omega_0(\mu'), \dots, \omega_{n_{sh}}(\mu'), X) \sim U(\Omega) \quad ,$$

and $s \in \mathcal{EVOP}(I. \mu, \mathcal{T}(I^2, [0, 1]) \times \mathcal{T}(I, \mathbb{R}). \Omega)$. Also. define $j : \{1, \dots . \mu'\} \times \mathcal{T}(I^2 . [0, 1]) \longrightarrow \{0. \dots . n_{sh}\}$ by

$$j(i, \theta) \quad \triangleq \quad \begin{cases} 0 & . \textit{ if } (\forall k)[X_{ik} > \theta(P_{\omega_0(i)}, P_{\omega_k(i)})] \\ \min\{k : X_{ik} \leq \theta(P_{\omega_0(i)} . P_{\omega_k(i)})\} & , \textit{ otherwise} \end{cases}$$

If for every $\theta : I^2 \longrightarrow [0. 1]$ (the threshold mapping*). every fitness function $\Phi : I \longrightarrow \mathbb{R}$. and every population $P \in I^\mu$. s satisfies*

$$[s_{(\theta,\Phi)}(P)]_i \quad = \quad \begin{cases} P_{\omega_0(i)} & , \textit{ if } \Phi(P_{\omega_0(i)}) \geq \Phi(P_{\omega_{j(i,\theta)}(i)}) \\ P_{\omega_{j(i,\theta)}(i)} & , \textit{ otherwise} \end{cases} \quad .$$

then s is called a binary tournament selection with thresholding operator. *If for every $\mathbf{a} \in I(\lambda_a)$ and every $\mathbf{b} \in I(\lambda_b)$. θ also satisfies $\lambda_a \neq \lambda_b \Longrightarrow \theta(\mathbf{a}, \mathbf{b}) = \delta_{\lambda,\ell}$, where δ is the Kronecker delta function, then s is called* length preserving. □

As in the deterministic case, the BTS with probabilistic thresholding operator is an order-based selection operator. Thus. it is strictly invariant (Theorem 2.5.5). Consequently. the effectiveness of any evolutionary algorithm using BTS with probabilistic thresholding, including the gfmGA, is unaffected by the choice of (order-preserving) fitness scaling function.

3.3 Algorithmic Specification

The preceding sections describe the novel operators used by the gfmGA. This section specifies the gfmGA in the formal framework of Section 2.3. and shows that the fmGA is a special case of the gfmGA.

Definition 3.3.1 (Generalized fast messy genetic algorithm): *Let*

- *I be an lfGA individual space over the genic alphabet $\{0,1\}$ with nominal string length ℓ and overflow factor o,*
- *$I(\lambda)$ defined by Equation 2,*
- $k \in \{1, \ldots, \ell\}$ *(the* building block size*)*,
- $t_f \in \mathbb{Z}^+$ *(the* final generation*)*,
- $t_p \in \{0, \ldots, t_f\}$ *(the* final primordial phase generation*)*,
- $\lambda^{(0)} \triangleq \ell - k$ *(the* initial individual length*)*.
- ψ *a sequence* $\{\psi^{(t)}\}_{t=0}^{t_p} \subset [0,1]^\ell$ *satisfying Equations 6 (the* filtering parameters*)*,
- $\alpha \in [0,1]$ *(the* probability of selection error*)*,
- $z_\alpha \in \mathbb{R}$ *such that* $Z \sim N(0,1) \Longrightarrow \Pr[Z \geq z_\alpha] = 1 - \alpha$,
- $\beta^2 \in \mathbb{R}^+$ *(the maximum* inverse signal-to-noise ratio per subfunction *to be detected)*,
- $\mu = \mu' \triangleq \frac{\ell!(\ell-2k)!}{(\ell-k)!^2} 2 z_\alpha^2 \beta^2 (\lceil \frac{\ell}{k} \rceil - 1) 2^k$ *(the* population size*)*.
- $\mathrm{c} \in I_F \triangleq I(\ell)$ *(the* competitive template),
- $\Phi_{\mathbf{c}} : I \longrightarrow \mathbb{R}$ *is an lfGA fitness function,*

- $\iota : \bigcup_{i=1}^{\infty} (I^{\mu})^i \longrightarrow \{\texttt{true}, \texttt{false}\}$ *(the termination criterion) such that*

$$\iota(\{P(0), \ldots, P(t)\}) = \texttt{true} \iff card(\{P(0), \ldots, P(t)\}) > t_f \quad ,$$

- r *a sequence* $\{r^{(t)}\}$ *of Goldberg's cut-and-splice operators* $r^{(t)} : \mathbb{R}^2 \longrightarrow \mathcal{T}\left(\Omega_r^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu^{(t)}}\right)\right)$,
- m *a sequence* $\{m^{(t)}\}$ *of evolutionary operators,*
- *for* $0 \leq t < t_p$, $m^{(t)} : [0,1]^{\ell} \longrightarrow \mathcal{T}\left(\Omega_m^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu^{(t)}}\right)\right)$ *a probabilistic BBF operator.*
- *for* $t_p \leq t < t_f$, $m^{(t)}$ *an identity evolutionary operator.*
- s *a sequence* $\{s^{(t)}\}$ *of BTS with probabilistic thresholding operators*

$$s^{(t)} : \mathcal{T}(I^2, [0,1]) \times \mathcal{T}(I, \mathbb{R}) \longrightarrow \mathcal{T}\left(\Omega_s^{(t)}, \mathcal{T}\left(I^{\mu^{(t)}}, I^{\mu'^{(t)}}\right)\right) \quad ,$$

- $\Theta_m^{(t)} \triangleq \lambda^{(t)}$ *for* $0 \leq t < t_p$ *(the* filtering parameters*).*
- $p_c^{(t)} = p_s^{(t)} \triangleq 0$ *for* $0 \leq t < t_p$,
- $\Theta_r^{(t)} \triangleq (p_c^{(t)}, p_s^{(t)}) \in \mathbb{R}^2$ *for* $0 \leq t < t_f$ *(the* cut-and-splice parameters*). and*
- θ *a sequence* $\{\theta^{(t)}\}$ *of threshold mappings* $\theta^{(t)} : I^2 \longrightarrow [0,1]$.

Then the algorithm shown in Figure 18 is called a fast messy genetic algorithm. □

```
$t := 0$;
initialize $P(0) := \{\mathbf{a}_1(0), \ldots, \mathbf{a}_{\mu^{(0)}}(0)\} \sim U(I(\lambda^{(0)}))$;
while $(\iota(\{P(0), \ldots, P(t)\}) \neq \texttt{true})$ do
        recombine: $P'(t) := r_{\Theta_r^{(t)}}(P(t))$;
        mutate: $P''(t) := m_{\Theta_m^{(t)}}(P'(t))$;
        select: $P(t+1) := s_{(\theta, \Phi_c)}(P''(t))$;
        $t := t + 1$;
od
```

Figure 18. Outline of a Generalized Fast Messy Genetic Algorithm

Building on the recommendation of Goldberg. et al. regarding the mGA [36], the algorithm may be applied iteratively for $2 \leq k \leq k_{max}$. The objective of iteration k is to identify an order-k optimal individual, given an order-$(k-1)$ optimal competitive template $\mathbf{c}$. The optimality condition on $\mathbf{c}$ for the second iteration $(k = 2)$ is order-1, which may be satisfied efficiently by hill-climbing in $I(\ell)$.

The gfmGA uses an order-invariant representation and a strictly invariant selection operator. Thus, it is a linkage-friendly genetic algorithm. The remainder of this section shows formally that the fmGA is a special case of the gfmGA. in the sense that every instantiation of the fmGA is equivalent to an instantiation of the gfmGA. but the converse does not hold. The first lemma considers the relationship between the deterministic and probabilistic BBF operators.

Lemma 3.3.2 *The building block filtering operator is a special case of the probabilistic building block filtering operator.*

Proof: Let I be an lfGA individual space. $\mu \in \mathbb{Z}$. $P \in I^{\mu}$. m the BBF operator, and $\lambda_f \in \{0, \ldots, \ell\}$. Then the probability that $[m_{\lambda_f}(P)]_i$ is of length λ is 1 if $\lambda = \lambda_f$ and 0 otherwise. This is equivalent to the probabilistic BBF operator with filtering parameters $\psi(\lambda) = \delta_{\lambda,\lambda_f}$. where δ is the Kronecker delta function. Now let ψ be any filtering parameters which are not of this form. and m the probabilistic BBF operator. Then with nonzero probability $[m_{\psi}(P)]_i$ is of length different than λ_f. ■

The next lemma considers the relationship between the deterministic and probabilistic BTS with thresholding operators.

Lemma 3.3.3 *The binary tournament selection with thresholding operator is a special case of the binary tournament selection with probabilistic thresholding operator.*

Proof: Let I be a non-empty set. $\mu \in \mathbb{Z}^+$, $P \in I^{\mu}$. s the BTS with thresholding operator. $d : I^2 \longrightarrow \mathrm{N}$. and $\theta : I^2 \longrightarrow \mathrm{N}$. Then the probability that $\mathbf{a}$ and $\mathbf{b}$ are θ-compatible is 1 if $d(\mathbf{a}, \mathbf{b}) \geq \theta(\mathbf{a}, \mathbf{b})$ and 0 otherwise.

This is equivalent to the BTS with probabilistic thresholding operator with threshold mapping

$$\Upsilon(\mathbf{a},\mathbf{b}) = \begin{cases} 1 & , \text{if } d(\mathbf{a},\mathbf{b}) \geq \theta(\mathbf{a},\mathbf{b}) \\ 0 & \text{if } d(\mathbf{a},\mathbf{b}) < \theta(\mathbf{a},\mathbf{b}) \end{cases} .$$

Now let Υ be any threshold mapping which is not of this form, and s the BTS with probabilistic thresholding operator. Then there exist individuals **a** and **b** with probability of Υ-compatibility $p \notin \{0, 1\}$. ■

The preceding lemmas imply that the fmGA is a special case of the gfmGA.

Theorem 3.3.4 *The fmGA is a special case of the set of the length-preserving gfmGA.*

Proof: The result follows immediately from Lemmas 3.3.2 and 3.3.3. the definitions of the fmGA and the gfmGA, and the observation that all individuals in an fmGA primordial phase population are of the same length. ■

This result implies the existence of probabilistic filtering and thresholding parameters for which the gfmGA expected effectiveness is no worse than the best possible fmGA expected effectiveness.

3.4 Summary

The generalized fast messy genetic algorithm (gfmGA) is a novel linkage-friendly genetic algorithm (lfGA). It shares the representation scheme of the messy genetic algorithm (mGA) and the fast messy genetic algorithm (fmGA), both of which are also lfGAs. The gfmGA also shares the high-level structure of the fmGA. The gemGA is another lfGA mentioned in Chapter II (see Figure 19). The gfmGA differs from the fmGA in its use of novel probabilistic generalizations of the building block filtering and binary tournament selection with thresholding operators. Because the fmGA is a special case of the gfmGA, existence is guaranteed of parameters for which the gfmGA expected effectiveness is no worse than the best possible fmGA expected effectiveness.

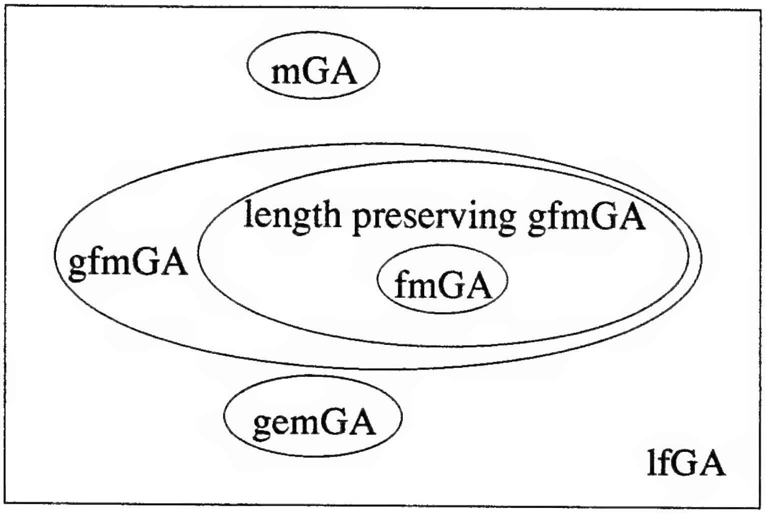


Figure 19. Venn Diagram of the Class of Linkage-Friendly Genetic Algorithms

IV. Probabilistic Building Block Filtering

The deterministic building block filtering (BBF) operators used by fast messy genetic algorithms delete the same fixed number of genes from each individual in a particular generation (Section 2.6.4). In contrast, the probabilistic BBF operators used by generalized fast messy genetic algorithms delete (or add) a random number of genes, the number being determined independently for each individual (Section 3.1). This chapter develops a dynamical systems model of probabilistic BBF which treats individuals as belonging to one of $2(\ell + 1)$ classes, where ℓ is the nominal string length. It is applicable in general to evolutionary algorithms with linkage-friendly genetic algorithm (lfGA) individual spaces over finite genic alphabets and to generalized fast messy genetic algorithms (Chapter III) in particular.

Deterministic BBF operators are modeled by Goldberg, et al. [35]. The model developed therein views individuals as belonging to one of two classes: those containing a particular building block, and those lacking it. The analysis is restricted to filtering operators which are purely destructive, as well as non-probabilistic in the sense that all individuals in a particular generation are of the same length. Kargupta extends the analysis of destructive non-probabilistic filtering operators to simultaneously consider multiple building blocks [47]. His analysis assumes that no individual contains more than one building block.

Together with the binary tournament selection model developed in Chapter V, the probabilistic BBF model permits the prediction of expected effectiveness. The prediction of expected effectiveness serves as the foundation for the exogenous parameter selection techniques proposed in Chapter VI.

After introducing the overall form of the mathematical model and certain notation (Section 4.1), the probabilities of building block survival (Section 4.2) and building block construction (Section 4.3) are developed. The total probability of building block presence after filtering is developed in Section 4.4.

4.1 Preliminaries

4.1.1 Notation. It is convenient to present certain definitions. Separable fitness functions are formally defined in terms of projection mappings.[1]

Definition 4.1.1 (Projection mapping): *Let I be an lfGA individual space with genic alphabet $\mathcal{A}$ and nominal string length ℓ, and $\mathcal{L} \triangleq \{1, \ldots, \ell\}$. If $\mathcal{P}_{\{L_1, \ldots, L_k\}} : \mathcal{A}^\ell \longrightarrow \mathcal{A}^k$ such that*

$$\mathcal{P}_{\{L_1, \ldots, L_k\}}(a_1, \ldots, a_\ell) \quad \triangleq \quad (a_{L_1}, \ldots, a_{L_k})$$

then $\mathcal{P}_{\{L_1, \ldots, L_k\}}$ is a projection mapping. □

When the set of loci $\mathcal{L}$ possesses a partition $\{\mathcal{L}_1, \ldots, \mathcal{L}_m\}$, the projection mappings $\mathcal{P}_{\mathcal{L}_i}$ may be thought of as "separating" the allele vector space $\mathcal{A}^\ell$ into independent smaller dimensional spaces $\mathcal{A}^{\mathrm{card}(\mathcal{L}_i)}$. When a fitness function Φ can be written as the sum of subfunctions operating on these independent spaces, Φ is *separable.*

Definition 4.1.2 (Order-k separable lfGA fitness function): *Let I be an lfGA individual space over genic alphabet $\mathcal{A}$ with nominal string length ℓ, and $\Phi_{\mathbf{c}}(\cdot) \triangleq T \circ f \circ D \circ \Gamma(\cdot, \mathbf{c}) : I \longrightarrow \mathbb{R}$ an lfGA fitness function. Suppose that for some fixed $k < \ell$ there exist*

- *a partition $\{\mathcal{L}_1, \ldots, \mathcal{L}_m\}$ of $\mathcal{L} \triangleq \{1, \ldots, \ell\}$ with each $k_i \triangleq card(\mathcal{L}_i) \leq k$; and*
- *functions $D_i : \mathcal{A}^{k_i} \longrightarrow \mathbb{R}^{n_i}$, $f_i : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}$, and $T_i : \mathbb{R} \longrightarrow \mathbb{R}$*

such that

$$\Phi_{\mathbf{c}}(\cdot) \quad = \quad \sum_{i=1}^{m} T_i(f_i(D_i(\mathcal{P}_{\mathcal{L}_i}(\Gamma(\cdot, \mathbf{c}))))) \quad ,$$

[1] The mappings defined are *not* "projection" operators in the sense of linear operator theory [61]. In particular, their domains and codomains are not in general the same space.

where the $\mathcal{P}_{\mathcal{L}_i}$*'s are projection mappings. Then* Φ_c *is called an* order-k separable lfGA fitness function. *Each* $\phi_i \triangleq T_i \circ f_i \circ D_i \circ \mathcal{P}_{\mathcal{L}_i} \circ \Gamma$ *is called a* subfunction. □

The notations $I(\lambda)$ and I_F for important subsets of the individual space I are introduced in Section 2.6.1. It is convenient to introduce special notation for other frequently mentioned subsets of I, which simplifies the analysis presented in the sequel.

- For each $\lambda \in \{0, \ldots, \ell\}$ and each $\beta \in \{1, \ldots, m\}$, assume there exists a unique order-k optimal building block $\{(a_1, l_1), \ldots, (a_{k_\beta}, l_{k_\beta})\}$, where $\mathcal{L}_\beta = \{l_1, \ldots, l_{k_\beta}\}$ is the set of defining loci for subfunction β, and define

$$I_\beta(\lambda) \triangleq \{(\mathbf{a}, \mathbf{l}) \in I(\lambda) : (\forall L \in \mathcal{L}_\beta)(\exists j \in \mathcal{L})[(a_j, l_j) = (a_L^*, L)]\} \ .$$

  Then $I_\beta(\lambda)$ is the set of length λ individuals which *contain building block* β.

- For each $\lambda \in \{0, \ldots, \ell\}$ and each $\beta \in \{1, \ldots, m\}$, define

$$I_{\neg\beta}(\lambda) \triangleq I(\lambda) - I_\beta(\lambda) \ .$$

  Then $I_{\neg\beta}(\lambda)$ is the set of length λ individuals which *lack building block* β.

Also, the set of primordial phase individuals containing building block β as

$$I_\beta \triangleq \bigcup_{\lambda=0}^{\ell} I_\beta(\lambda) \ .$$

Finally, the presentation is notationally simplified through the use of the hypergeometric probability density function. If X is a random variable with a hypergeometric distribution [53], then

$$\Pr[X = x] = h(x; n, M, N) \triangleq \frac{\binom{M}{x}\binom{N-M}{n-x}}{\binom{N}{n}} \ . \tag{8}$$

4.1.2 Dynamical Systems Model. Liepins and Vose [49], propose a dynamical systems view of a *population vector* p as a probability density function over the individual space, with the ith component being the probability with which the ith individual is sampled. This research views the population vector more generally as a density function over a set of equivalence classes which form a partition of the individual space. Specifically, the population vector is of the form

$$\mathbf{p} \triangleq (p_0, p_1, \ldots, p_\ell \mid q_0, q_1, \ldots, q_\ell)^\top \ .$$

Each component p_i is the probability that an individual sampled from the population belongs to the equivalence class $I_\beta(i)$ consisting of those individuals which contain building block β and have length i.[2] Similarly, each component q_i is the probability that an individual sampled from the population belongs to the equivalence class $I_{\neg\beta}(i)$ consisting of those individuals which lack building block β and have length i.

Probabilistic BBF is modeled as a deterministic transition function τ_m mapping the current population vector $\mathbf{p}$ to the expected next population vector $\tau_m(\mathbf{p})$. Because the next population vector in an infinite population algorithm exactly matches the expected population vector, the model developed here is exact for such algorithms. Furthermore, the transition function τ_m is independent of population size (see Vose and Wright [74]). Hence, the model is also an exact expected value model for finite population size algorithms.

4.2 Building Block Survival

This section develops a mathematical model of building block survival under the probabilistic BBF operator. The *probability of building block survival* is the conditional probability that an individual contains the building block after filtering given that it does before filtering. This probability depends on the length of the individual before and after filtering, as stated in the following theorem:

[2] Strictly speaking, building blocks are defined only in the case of separable fitness functions. For this reason, the theorems in this chapter are proved in the context of such fitness functions. The results apply also to fitness functions which are not separable, with the understanding that the "building block" is simply the globally optimal individual.

Theorem 4.2.1 (Building block survival) *Let I be an lfGA individual space with nominal string length ℓ, $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ a separable fitness function. $\mathcal{L}_\beta$ the set of defining loci of ϕ_β, $k \triangleq card\,(\mathcal{L}_\beta)$, and m' a local probabilistic BBF operator. If $\mathbf{X} \in I(\lambda)$ and $m'(\mathbf{X}) \in I(\tilde{\lambda})$ then the probability of building block survival is*

$$p_{surv}(k, \tilde{\lambda}. \lambda, \ell) \triangleq \begin{cases} h(k; \tilde{\lambda}, k. \lambda) & , \text{ if } 0 \leq \tilde{\lambda} < \lambda \\ 1 & , \text{ if } \lambda \leq \tilde{\lambda} \leq \ell \end{cases} , \qquad (9)$$

where h is defined by Equation 8.

Proof: Consider first the case $\lambda \leq \tilde{\lambda} \leq \ell$. No genes are deleted by m', so the building block survives with probability 1. Now consider the case $0 \leq \tilde{\lambda} < k$. Then $m'(\mathbf{X})$ does not contain enough genes to contain building block β, so the building block survives with probability $0 = h(k: \tilde{\lambda}, k, \lambda)$. Finally, consider the case $k \leq \tilde{\lambda} < \lambda$. Then there are $\binom{\lambda}{\tilde{\lambda}}$ ways to choose the $\tilde{\lambda}$ genes to keep from the original λ. Also, there are $\binom{k}{k}\binom{\lambda-k}{\tilde{\lambda}-k}$ ways to choose all k genes of the building block and $\tilde{\lambda} - k$ more genes from the remaining $\lambda - k$ so that the building block survives. Thus, for the case $k \leq \tilde{\lambda} < \lambda$, the probability that filtering does not disrupt an existing building block is

$$\Pr[m'(\mathbf{X}) \in I_\beta \mid \mathbf{X} \in I_\beta(\lambda) \wedge m'(\mathbf{X}) \in I(\tilde{\lambda})] = \frac{\binom{k}{k}\binom{\lambda-k}{\tilde{\lambda}-k}}{\binom{\lambda}{\tilde{\lambda}}} = h(k; \tilde{\lambda}, k, \lambda) ,$$

which completes the proof. ■

Theorem 4.2.1 is essentially a re-statement of the probability of survival claimed by Goldberg et al. [35] and later by Kargupta (see Section 2.6.5.2) for a deterministic BBF operator. generalized to apply to probabilistic BBF operators. The following theorem gives the total probability that an individual contains a particular building block both before and after filtering. and that it is of particular lengths before and after filtering:

Theorem 4.2.2 (Building block presence before and after filtering)

Let I be an lfGA individual space with nominal string length ℓ, $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ a separable fitness function, $\mathcal{L}_\beta$ the set of defining loci of ϕ_β, $k \triangleq card\,(\mathcal{L}_\beta)$. and m' a local probabilistic BBF operator. If $\mathbf{X}$ is *drawn randomly from population $P(t)$, then*

$$\Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \;=\; p_{surv}(k, \tilde{\lambda}, \lambda, \ell) \cdot \psi(\tilde{\lambda}, t) \cdot \Pr[\mathbf{X} \in I_\beta(\lambda)] \quad , \qquad (10)$$

where ψ is the filtering schedule, and the $p_{surv}(k, \tilde{\lambda}, \lambda, \ell)$'s are defined by Equation 9.

Proof: The event $m'(\mathbf{X}) \in I_\beta(\tilde{\lambda})$ is equivalent to the event $m'(\mathbf{X}) \in I_\beta \wedge m'(\mathbf{X}) \in I(\tilde{\lambda})$, so the probability on the left hand side of Equation 10 may be written as

$$\Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \;=\; \Pr[m'(\mathbf{X}) \in I_\beta \wedge m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \quad .$$

By the definition of probabilistic BBF operator, the event $m'(\mathbf{X}) \in I(\tilde{\lambda})$ is independent of the event $\mathbf{X} \in I_\beta(\lambda)$, i.e.

$$\begin{aligned} \Pr[m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] &= \Pr[m'(\mathbf{X}) \in I(\tilde{\lambda})] \cdot \Pr[\mathbf{X} \in I_\beta(\lambda)] \\ &= \psi(\tilde{\lambda}, t) \cdot \Pr[\mathbf{X} \in I_\beta(\lambda)] \quad . \end{aligned}$$

Suppose $\Pr[m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] = 0$. Then

$$\begin{aligned} \Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] &= 0 \\ &= p_{surv}(k, \tilde{\lambda}, \lambda, \ell) \cdot 0 \\ &= p_{surv}(k, \tilde{\lambda}, \lambda, \ell) \cdot \psi(\tilde{\lambda}, t) \cdot \Pr[\mathbf{X} \in I_\beta(\lambda)] \quad . \end{aligned}$$

On the other hand, if $\Pr[m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \neq 0$ then

$$\begin{aligned}
&\Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \\
&\quad = \Pr[m'(\mathbf{X}) \in I_\beta \mid m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \cdot \Pr[m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \\
&\quad = \Pr[m'(\mathbf{X}) \in I_\beta \mid m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] \cdot \psi(\tilde{\lambda}, t) \cdot \Pr[\mathbf{X} \in I_\beta(\lambda)] \ .
\end{aligned}$$

Because $\Pr[m'(\mathbf{X}) \in I_\beta \mid m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)]$ is the probability of building block survival. the result follows immediately from Theorem 4.2.1. ■

4.3 Building Block Construction

The deterministic BBF operators of fast messy genetic algorithms are such that all individuals in generation $t+1$ have lengths no greater than those in generation t. Probabilistic BBF operators do not necessarily exhibit this property. Thus. it is possible for a probabilistic BBF operator to construct building blocks as well as disrupt them. This section develops a mathematical model of building block construction in generalized fast messy genetic algorithms.

The *probability of building block construction* is the conditional probability that an individual contains a particular building block β after filtering given that it lacks the building block before filtering. This probability is nonzero only when the unfiltered individual contains no incorrect genes. When this condition holds, the probability depends on the length of the individual both before and after filtering. as well as the number of missing genes. The number of missing genes in an individual $\mathbf{X}$. i.e. the number of loci of subfunction β with respect to which $\mathbf{X}$ is unspecified. is denoted $\tilde{K}(\mathbf{X})$. That is. $\tilde{K}(\mathbf{X}) = \tilde{k}$ if and only if $\mathbf{X}$ is specified (correctly or otherwise) with respect to exactly $k - \tilde{k}$ of the loci of subfunction β. where $k \triangleq \operatorname{card}(\mathcal{L}_\beta)$ and $\mathcal{L}_\beta$ is the set of defining loci for subfunction β. The relationship between the probability of building block construction. the length of the individual before and after filtering. and the number of missing genes is given by the following theorem:

Theorem 4.3.1 (Building block construction) *Let I be an lfGA individual space over finite genic alphabet $\mathcal{A}$ with nominal string length ℓ, $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ a separable fitness function, $\mathcal{L}_\beta$ the set of defining loci of ϕ_β, $k \triangleq card\,(\mathcal{L}_\beta)$, and m' a local probabilistic BBF operator. If the alleles of pre-existing genes are correct with probability $[card\,(\mathcal{A})]^{-1}$, then the conditional probability of building block construction given that $\tilde{k}$ genes are missing is*

$$
\begin{aligned}
&\Pr[m'(\mathbf{X}) \in I_\beta \mid m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \\
&= \begin{cases} 0 & , \mathit{if}\ 0 \le \tilde{\lambda} < \lambda + \tilde{k} \\ [card\,(\mathcal{A})]^{-k} h(\tilde{k}; \tilde{\lambda} - \lambda, \tilde{k}, \ell - \lambda) & , \mathit{if}\ \lambda + \tilde{k} \le \tilde{\lambda} \le \ell \end{cases}
\end{aligned}
$$

Proof: Suppose first $0 \le \tilde{\lambda} < \lambda + \tilde{k}$. Then the filtering operator does not add enough genes to complete building block β, so the building block is constructed with probability 0. Now suppose that $\lambda + \tilde{k} \le \tilde{\lambda} \le \ell$. Then there are $\binom{\ell-\lambda}{\tilde{\lambda}-\lambda}$ ways to choose $\tilde{\lambda} - \lambda$ loci to specify from the $\ell - \lambda$ available. Also, there are $\binom{\tilde{k}}{\tilde{k}}\binom{\ell-\lambda-\tilde{k}}{\tilde{\lambda}-\lambda-\tilde{k}}$ ways to choose the $\tilde{k}$ missing loci of subfunction β, as well as $(\tilde{\lambda}-\lambda)-\tilde{k}$ more loci from the remaining $(\ell-\lambda)-\tilde{k}$. The alleles of the $\tilde{k}$ new genes are correct with (independent) probability $[\text{card}\,(\mathcal{A})]^{-1}$. By hypothesis, the alleles of the $k - \tilde{k}$ pre-existing genes are also correct with (independent) probability $[\text{card}\,(\mathcal{A})]^{-1}$. Thus, the probability of building block construction is

$$
\begin{aligned}
\Pr[m'(\mathbf{X}) \in I_\beta \mid \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] &= [\text{card}\,(\mathcal{A})]^{-(\tilde{k}+k-\tilde{k})} \frac{\binom{\tilde{k}}{\tilde{k}}\binom{\ell-\lambda-\tilde{k}}{\tilde{\lambda}-\lambda-\tilde{k}}}{\binom{\ell-\lambda}{\tilde{\lambda}-\lambda}} \\
&= [\text{card}\,(\mathcal{A})]^{-k} h(\tilde{k}; \tilde{\lambda} - \lambda, \tilde{k}, \ell - \lambda) \quad .
\end{aligned}
$$

which completes the proof. ■

Theorem 4.3.1 is the building block construction analogue of Theorem 4.2.1. The following lemma is useful in the proof of the analogue of Theorem 4.2.2.

Lemma 4.3.2 *Let I be an lfGA individual space over genic alphabet $\mathcal{A}$ with nominal string length ℓ such that $1 < card\,(\mathcal{A}) < \infty$, $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ a separable fitness function, $\mathcal{L}_\beta$ the set of defining loci of ϕ_β, and $k \triangleq card\,(\mathcal{L}_\beta)$. If $\mathbf{X}$ is randomly drawn from population $P(t)$ such that*

$$\mathbf{x} \in I(\lambda) \Longleftrightarrow \Pr[\mathbf{X} = \mathbf{x}] = \psi(\lambda; t-1) \cdot [N(\lambda)]^{-1} \quad , \tag{11}$$

where ψ is the filtering schedule and $N(\lambda)$ is defined in Section 5.3.1, then for each $\tilde{k} \in \{1, \ldots, k\}$

$$\Pr[\mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] =$$
$$[1 - [card\,(\mathcal{A})]^{-k} h(k; \lambda, k, \ell)]^{-1} \cdot \left\{ \psi(\lambda; t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda)] \right\} \cdot h(k - \tilde{k}; \lambda, k, \ell) \quad .$$

Proof: Because $I_\beta(\lambda)$ and $I_{\neg\beta}(\lambda)$ form a partition of $I(\lambda)$,

$$\begin{aligned} \Pr[\mathbf{X} \in I_{\neg\beta}(\lambda)] &= \Pr[\mathbf{X} \in I(\lambda)] - \Pr[\mathbf{X} \in I_\beta(\lambda)] \\ &= \psi(\lambda; t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda)] \quad . \end{aligned}$$

Suppose that $\Pr[\mathbf{X} \in I_{\neg\beta}(\lambda)] = 0$. Then

$$\begin{aligned} &\Pr[\mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \\ &= 0 \\ &= [1 - [\mathrm{card}\,(\mathcal{A})]^{-k} h(k; \lambda. k. \ell)]^{-1} \cdot 0 \cdot h(k - \tilde{k}; \lambda, k, \ell) \\ &= [1 - [\mathrm{card}\,(\mathcal{A})]^{-k} h(k; \lambda. k. \ell)]^{-1} \cdot \left\{ \psi(\lambda; t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda)] \right\} \cdot h(k - \tilde{k}; \lambda. k, \ell) \quad . \end{aligned}$$

On the other hand, if $\Pr[\mathbf{X} \in I_{\neg\beta}(\lambda)] \neq 0$ then

$$\begin{aligned}
\Pr[\mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] &= \Pr[\mathbf{X} \in I_{\neg\beta}(\lambda)] \cdot \Pr[\tilde{K}(\mathbf{X}) = \tilde{k} \mid \mathbf{X} \in I_{\neg\beta}(\lambda)] \\
&= \{\psi(\lambda; t-1) - \Pr[\mathbf{X} \in I_{\beta}(\lambda)]\} \cdot \Pr[\tilde{K}(\mathbf{X}) = \tilde{k} \mid \mathbf{X} \in I_{\neg\beta}(\lambda)] \quad .
\end{aligned}$$

By the definition of a conditional probability,

$$\Pr[\tilde{K}(\mathbf{X}) = \tilde{k} \mid \mathbf{X} \in I_{\neg\beta}(\lambda)] \triangleq \frac{\Pr[\tilde{K}(\mathbf{X}) = \tilde{k} \wedge \mathbf{X} \in I_{\neg\beta}(\lambda)]}{\Pr[\mathbf{X} \in I_{\neg\beta}(\lambda)]} \quad .$$

For $\mathbf{X}$ sampled according to Equation 11 and $\tilde{k} > 0$,

$$\begin{aligned}
\Pr[\tilde{K}(\mathbf{X}) = \tilde{k} \mid \mathbf{X} \in I_{\neg\beta}(\lambda)] &= \frac{\mathrm{card}\left(\{\mathbf{X} \in I_{\neg\beta}(\lambda) : \tilde{K}(\mathbf{X}) = \tilde{k}\}\right)}{N_{\neg\beta}(\lambda)} \\
&= \frac{[\mathrm{card}\,(\mathcal{A})]^{\lambda} \binom{k}{k-\tilde{k}} \binom{\ell-k}{\lambda-(k-\tilde{k})}}{[\mathrm{card}\,(\mathcal{A})]^{\lambda} \binom{\ell}{\lambda} - [\mathrm{card}\,(\mathcal{A})]^{\lambda-k} \binom{\ell-k}{\lambda-k}} \\
&= \left[\frac{\binom{\ell}{\lambda} - [\mathrm{card}\,(\mathcal{A})]^{-k} \binom{\ell-k}{\lambda-k}}{\binom{k}{k-\tilde{k}} \binom{\ell-k}{\lambda-(k-\tilde{k})}} \right]^{-1} \\
&= \left\{ [h(k-\tilde{k}; \lambda, k, \ell)]^{-1} \left[1 - [\mathrm{card}\,(\mathcal{A})]^{-k} \frac{\binom{\ell-k}{\lambda-k}}{\binom{\ell}{\lambda}} \right] \right\}^{-1} \\
&= h(k-\tilde{k}: \lambda, k, \ell) \left[1 - [\mathrm{card}\,(\mathcal{A})]^{-k} h(k: \lambda, k, \ell) \right]^{-1} \quad ,
\end{aligned}$$

which completes the proof. ■

The final theorem of this section gives the total probability that an individual lacks a particular building block before filtering, contains the building block after filtering, and is of particular lengths before and after filtering.

Theorem 4.3.3 (Building block presence only after filtering) *Let I be an lfGA individual space over genic alphabet $\mathcal{A}$ with nominal string length ℓ such that $1 < card(\mathcal{A}) < \infty$, $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ a separable fitness function, $\mathcal{L}_\beta$ the set of defining loci of ϕ_β, and $k \triangleq card(\mathcal{L}_\beta)$. If $\mathbf{X}$ is randomly drawn from* *population $P(t)$ according to Equation 11 then*

$$\Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda)] \quad = \quad \psi(\tilde{\lambda};t) \cdot \left\{\psi(\lambda;t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda)\right\} \cdot p_{cons}(k,\tilde{\lambda}.\lambda.\ell), \quad .$$

where

$$p_{cons}(k.\tilde{\lambda}.\lambda.\ell) \triangleq \begin{cases} 0 & , \textit{if } 0 \leq \tilde{\lambda} < \lambda + \tilde{k} \\ [[\mathit{card}\,(\mathcal{A})]^{-k} - h(k:\lambda,k.\ell)]^{-1} \sum_{\tilde{k}=1}^{k} h(k-\tilde{k};\lambda.k.\ell) \cdot h(\tilde{k}.\tilde{\lambda}-\lambda,\tilde{k}.\ell-\lambda) & , \textit{if } \lambda + \tilde{k} \leq \tilde{\lambda} \leq \ell \end{cases} .$$

Proof: By the definition of a local BBF operator, the event $m'(\mathbf{X}) \in I(\tilde{\lambda})$ is independent of the event $\mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}$, i.e.

$$\begin{aligned} &\Pr[m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \\ &= \quad \Pr[m'(\mathbf{X}) \in I(\tilde{\lambda})] \cdot \Pr[\mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \\ &= \quad [1 - [\text{card}\,(\mathcal{A})]^{-k} h(k;\lambda.k,\ell)]^{-1} \psi(\tilde{\lambda},t) \cdot \left\{\psi(\lambda;t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda)]\right\} \cdot h(k-\tilde{k};\lambda.k.\ell) \quad , \end{aligned}$$

where we have used Lemma 4.3.2. By Theorem 4.3.1.

$$\begin{aligned} &\Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \\ &= \quad \Pr[m'(\mathbf{X}) \in I_\beta \wedge m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \\ &= \quad \Pr[m'(\mathbf{X}) \in I_\beta \mid m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \\ &\qquad \cdot \Pr[m'(\mathbf{X}) \in I(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \end{aligned}$$

$$= \begin{cases} 0. & \text{if } 0 \le \tilde{\lambda} < \lambda + \tilde{k} \\ [\text{card}\,(\mathcal{A})]^{-k} h(\tilde{k}; \tilde{\lambda} - \lambda, \tilde{k}, \ell - \lambda) \cdot [1 - [\text{card}\,(\mathcal{A})]^{-k} h(k; \lambda, k, \ell)]^{-1} & \\ \cdot\psi(\tilde{\lambda}, t) \cdot \left\{\psi(\lambda; t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda)]\right\} \cdot h(k - \tilde{k}; \lambda, k, \ell). & \text{if } \lambda + \tilde{k} \le \tilde{\lambda} \le \ell \end{cases}$$

$$= \begin{cases} 0, & \text{if } 0 \le \tilde{\lambda} < \lambda + \tilde{k} \\ [[\text{card}\,(\mathcal{A})]^{-k} - h(k; \lambda, k, \ell)]^{-1} \cdot \psi(\tilde{\lambda}, t) \cdot \left\{\psi(\lambda; t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda)]\right\} & \\ \cdot h(\tilde{k}: \tilde{\lambda} - \lambda, \tilde{k}, \ell - \lambda) \cdot h(k - \tilde{k}: \lambda, k, \ell), & \text{if } \lambda + \tilde{k} \le \tilde{\lambda} \le \ell \end{cases}$$

Finally, by the Law of Total Probability and the observation that

$$\Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = 0] \quad = \quad 0 \ ,$$

it follows that

$$\Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda)] \quad = \quad \sum_{\tilde{k}=1}^{k} \Pr[m'(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda) \wedge \tilde{K}(\mathbf{X}) = \tilde{k}] \ ,$$

which completes the proof. ■

4.4 Total Probability of Building Block Presence

For a probabilistic BBF operator, individuals which are not of the same length before filtering may be after filtering. The following theorem gives the total probability that after filtering an individual is of length $\tilde{\lambda}$ and contains building block β.

Theorem 4.4.1 (Total probability of building block presence)

Let I be an lfGA individual space over genic alphabet $\mathcal{A}$ with nominal string length ℓ such that $1 < \text{card}\,(\mathcal{A}) < \infty$, $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ a separable fitness function, $\mathcal{L}_\beta$ the set of defining loci of ϕ_β, and $k \triangleq \text{card}\,(\mathcal{L}_\beta)$.

If **X** *is randomly drawn from population* $P(t)$ *according to Equation 11 then*

$$\begin{aligned}
&\Pr[m(\mathbf{X}) \in I_\beta(\tilde{\lambda})] \\
&= \psi(\tilde{\lambda}, t) \cdot \sum_{\lambda=0}^{\ell} \left(\Pr[\mathbf{X} \in I_\beta(\lambda)] \cdot p_{surv}(k, \tilde{\lambda}, \lambda) + \left\{ \dot{\psi}(\lambda; t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda) \right\} \cdot p_{cons}(k, \tilde{\lambda}, \lambda) \right) \quad .
\end{aligned}$$

Proof: The $I(\lambda)$'s form a partition of I. Furthermore, $I_\beta(\lambda)$ and $I_{\neg\beta}(\lambda)$ form a partition of $I(\lambda)$. Thus, by the Law of Total Probability, the probability that an individual contains building block β after filtering is

$$\begin{aligned}
&\Pr[m(\mathbf{X}) \in I_\beta(\tilde{\lambda})] \\
&= \Pr[m(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I] \\
&= \sum_{\lambda=0}^{\ell} \left\{ \Pr[m(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_\beta(\lambda)] + \Pr[m(\mathbf{X}) \in I_\beta(\tilde{\lambda}) \wedge \mathbf{X} \in I_{\neg\beta}(\lambda)] \right\} \\
&= \sum_{\lambda=0}^{\ell} \left(\psi(\tilde{\lambda}, t) \cdot \Pr[\mathbf{X} \in I_\beta(\lambda)] \cdot p_{surv}(k, \tilde{\lambda}, \lambda) + \psi(\tilde{\lambda}: t) \cdot \left\{ \psi(\lambda: t-1) - \Pr[\mathbf{X} \in I_\beta(\lambda) \right\} \cdot p_{cons}(k. \tilde{\lambda}. \lambda) \right)
\end{aligned}$$

where we have used Theorems 4.2.2 and 4.3.3. ■

4.5 Summary

The building block filtering (BBF) operators used in fast messy genetic algorithms delete the same fixed number of genes from each individual in a particular generation. The probabilistic BBF operators used in generalized messy genetic algorithms either add or delete a random number of genes, determined independently for each individual. Previous models are limited to deterministic BBF operators, and consequently consider only the probability of building block survival. The mathematical model developed here extends existing analysis of building block survival to the probabilistic case, and incorporates analysis of building block construction to arrive at the total probability of building block presence following filtering. Together with the binary tournament selection model developed in Chapter V, this model permits the prediction of the expected effectiveness given a particular choice of exogenous parameters. The prediction of expected effectiveness forms the basis for the parameter selection techniques proposed in Chapter VI.

V. Binary Tournament Selection with Probabilistic Thresholding

Previously proposed models of tournament selection (see Section 2.6.2.3) focus on either takeover time (e.g. Goldberg and Deb [33]) or on selection intensity (e.g. Bäck [5] and Blickle and Thiele [11]). Those models in the latter category treat all individuals as belonging to a single class, so that their fitnesses are independent and identically distributed. Neither class of models provides information regarding the relative growth of one (multiple individual) class of individuals with respect to another. Furthermore, no previous model of tournament selection considers thresholding.

This chapter develops a model of binary tournament selection (BTS) with probabilistic thresholding which treats individuals as belonging to one of two classes with possibly differing fitness distributions. The model is applicable to evolutionary algorithms using binary tournament selection with thresholding where the more fit individual is selected with probability 1. The models of binary tournament selection and probabilistic building block filtering (developed in Chapter IV) allow the prediction of expected effectiveness resulting from a choice of filtering and thresholding parameters. The prediction of expected effectiveness serves as the foundation for the exogenous parameter selection techniques proposed in Chapter VI.

One key component of the proposed tournament selection model is the probability of "correct decision making," defined and analyzed in Section 5.1. A dynamical systems model of BTS with probabilistic thresholding is developed in Section 5.2 using Markov chain analysis. The distribution after selection depends on the initial distribution. which is examined in Section 5.3 for the case of a uniformly distributed population in a linkage-friendly genetic algorithm. Finally, Section 5.4 discusses the application of the model to the prediction of building block processing in a fast messy genetic algorithm.

5.1 Decision Making

Each (nontrivial) tournament performed in the application of a binary tournament selection operator may be viewed as a decision between two classes of individuals $A, B \subset I$ where one competing individual belongs to A and the other to B.[1] This section develops a probabilistic model of "correct" decision making as a function of

- the fitness distributions F and G of those initial population individuals belonging to A and B, respectively;
- the number of ancestors which belong to A and B for each of the competing individuals;
- the individual similarity $D = d(\mathbf{a}, \mathbf{b})$ between the competing individuals $\mathbf{a} \in A$ and $\mathbf{b} \in B$ (typically the number of common defining loci); and
- the conditional fitness distributions F_Ω and G_Ω of those individuals belonging to A and B, respectively, given that the pair belongs to the set $\Omega(D)$ of such individual pairs also having individual similarity D.

Section 5.1.1 defines order statistics, upon which the model developed in this section is based. The section also extends the standard theory to obtain the distribution of the maximal order statistic of a set of random variables only some of which are identically distributed. This result is applied to derive the probabilities of correct decision making in the absence (Section 5.1.2), and in the presence (Section 5.1.3) of thresholding.

5.1.1 Order Statistics.

Arnold, et al. define order statistics as follows:

Definition 5.1.1 (Order statistic (Arnold, et al. [4])): *Suppose that $(X_1, \ldots, X_n)$ are n jointly distributed random variables. The corresponding order statistics are the X_i's arranged in nondecreasing order. The smallest of the X_i's is denoted by $X_{1:n}$, the second smallest is denoted by $X_{2:n}$, …, and, finally, the largest is denoted by $X_{n:n}$. Thus $X_{1:n} \leq X_{2:n} \leq \cdots \leq X_{n:n}$.* □

[1] It is convenient to think of A and B as forming a partition of the individual space I so that $A \cap B = \{\}$ and $A \cup B = I$, but only the disjointness condition is necessary to the decision making model developed here.

For the case that the X_i's are independent and identically distributed, it is well known that the distribution function of $X_{i:n}$ is

$$F_{i:n}(x) = \sum_{r=i}^{n} \binom{n}{r} [F(x)]^r [1 - F(x)]^{n-r} \quad .$$

and that the density function is

$$f_{1:n}(x) = \frac{n!}{(i-1)!(n-i)!} [F(x)]^{i-1} [1 - F(x)]^{n-i} f(x) \quad ,$$

where F is the distribution function of the X_i's and f is the density function. For many of the order statistics which appear in the subsequent analysis, the underlying random variables are not identically distributed, but are partitioned into two sets within each of which the random variables are identically distributed. Of particular interest is the conditional distribution of the maximal order statistic, given that its underlying random variable possesses a particular distribution. This statistic is shown in Sections 5.1.2 and 5.1.3 to be related to the fitness distributions of individuals surviving tournament selection.

Theorem 5.1.2 *Let $(X_1, \ldots, X_{n_X})$ be $n_X \geq 0$ identically distributed random variables with density function f and distribution function F. Also, let $(Y_1, \ldots, Y_{n_Y})$ be $n_Y \geq 0$ identically distributed random variables with density function g and distribution function G. Finally, let Z be a random variable with density function f_Ω. If the X_i's, the Y_i's, and Z are mutually independent, then the conditional distribution function of Z given that $n_X > 0 \Longrightarrow Z \geq X_{n_X:n_X}$ and $n_Y > 0 \Longrightarrow Z \geq Y_{n_Y:n_Y}$ is*

$$H(t) \quad = \quad K \int_{-\infty}^{t} f_\Omega(z) [F(z)]^{n_X} [G(z)]^{n_Y} \, dz \quad .$$

and the conditional density function is

$$h(t) \quad = \quad K f_\Omega(t) [F(t)]^{n_X} [G(t)]^{n_Y} \quad ,$$

where

$$K \triangleq \frac{1}{\int_{-\infty}^{\infty} f_\Omega(z)[F(z)]^{n_X}[G(z)]^{n_Y}\,dz} .$$

Proof: Suppose first that $n_X > 0$ and $n_Y > 0$. The joint density function of Z and the order statistics $X_{1:n_X}, \ldots, X_{n_X:n_X}$ and $Y_{1:n_Y}, \ldots, Y_{n_Y:n_Y}$ may be found by transformation of variables [41] and their mutual independence to be

$$h_{1,\ldots,n_X:n_X;1,\ldots,n_Y:n_Y;z}(x_1, \ldots, x_{n_X}, y_1, \ldots, y_{n_Y}, z) = \left[n_X! \prod_{r=1}^{n_X} f(x_r)\right]\left[n_Y! \prod_{s=1}^{n_Y} g(y_s)\right] f_\Omega(z) .$$

The joint density function of $X_{n_X:n_X}$, $Y_{n_Y:n_Y}$, and Z is found by "integrating out" the remaining variables, where the limits of integration are determined by the definitions of the order statistics:

$$\begin{aligned}
h_{n_X:n_X:n_Y:n_Y:z}(x_{n_X}, y_{n_Y}, z) &= n_X!n_Y!f(x_{n_X})g(y_{n_Y})f_\Omega(z) \\
&\quad \cdot \left[\int_{-\infty}^{x_{n_X}} \cdots \int_{-\infty}^{x_2} f(x_1)\cdots f(x_{n_X-1})\,dx_1 \cdots dx_{n_X-1}\right] \\
&\quad \cdot \left[\int_{-\infty}^{y_{n_Y}} \cdots \int_{-\infty}^{y_2} g(y_1)\cdots g(y_{n_Y})\,dy_1 \cdots dy_{n_Y-1}\right] \\
&= n_X!n_Y!f(x_{n_X})g(y_{n_Y})f_\Omega(z)\left[\frac{[F(x_{n_X})]^{n_X-1}}{(n_X-1)!}\right]\left[\frac{[G(y_{n_Y})]^{n_Y-1}}{(n_Y-1)!}\right] \\
&= n_X n_Y f(x_{n_X})g(y_{n_Y})f_\Omega(z)[F(x_{n_X})]^{n_X-1}[G(y_{n_Y})]^{n_Y-1} .
\end{aligned}$$

The conditional distribution function is thus

$$\begin{aligned}
H(t) &\triangleq \Pr[Z \leq t \mid Z \geq X_{n_X:n_X} \wedge Z \geq Y_{n_Y:n_Y}] \\
&= \frac{\Pr[Z \leq t \wedge Z \geq X_{n_X:n_X} \wedge Z \geq Y_{n_Y:n_Y}]}{\Pr[Z \geq X_{n_X:n_X} \wedge Z \geq Y_{n_Y:n_Y}]} \\
&= \frac{\int_{-\infty}^{t} \int_{-\infty}^{z} \int_{-\infty}^{z} h_{n_X:n_X;n_Y:n_Y;z}(x, y, z)\, dx\, dy\, dz}{\int_{-\infty}^{\infty} \int_{-\infty}^{z} \int_{-\infty}^{z} h_{n_X:n_X;n_Y:n_Y;z}(x, y, z)\, dx\, dy\, dz} \\
&= \frac{\int_{-\infty}^{t} f_\Omega(z)[F(z)]^{n_X}[G(z)]^{n_Y}\, dz}{\int_{-\infty}^{\infty} f_\Omega(z)[F(z)]^{n_X}[G(z)]^{n_Y}\, dz} \quad .
\end{aligned}$$

Now suppose that $n_X > 0$ and $n_Y = 0$. Then the joint density function of $X_{n_X:n_X}$ and Z is

$$h_{n_X:n_X:z}(x_{n_X}, z) \quad = \quad n_X f(x_{n_X}) f_\Omega(z)[F(x_{n_X})]^{n_X - 1} \quad .$$

and the conditional distribution function is

$$\begin{aligned}
H(t) &= \frac{\int_{-\infty}^{t} \int_{-\infty}^{z} h_{n_X:n_X:z}(x, z)\, dx\, dz}{\int_{-\infty}^{\infty} \int_{-\infty}^{z} h_{n_X:n_X;z}(x, z)\, dx\, dz} \\
&= \frac{\int_{-\infty}^{t} f_\Omega(z)[F(z)]^{n_X}\, dz}{\int_{-\infty}^{\infty} f_\Omega(z)[F(z)]^{n_X}\, dz} \\
&= \frac{\int_{-\infty}^{t} f_\Omega(z)[F(z)]^{n_X}[G(z)]^{n_Y}\, dz}{\int_{-\infty}^{\infty} f_\Omega(z)[F(z)]^{n_X}[G(z)]^{n_Y}\, dz} \quad .
\end{aligned}$$

Likewise, for $n_X = 0$ and $n_Y > 0$ the joint density function is

$$h_{n_Y:n_Y:z}(y_{n_Y}, z) \quad = \quad n_Y g(y_{n_Y}) f_\Omega(z)[G(y_{n_Y})]^{n_Y - 1} \quad .$$

and the same conditional distribution function is obtained. Finally, for $n_X = n_Y = 0$, the density function is just f_Ω, and again the same conditional distribution function is obtained. It remains only to note that the conditional density function h is the unique function satisfying $H(t) = \int_{-\infty}^{t} h(s)\, ds$. ∎

5.1.2 Decision Making With Trivial Thresholding. Define the random variables $X \triangleq \Phi_c(\mathbf{a})$ and $Y \triangleq \Phi_c(\mathbf{b})$ where $\mathbf{a} \in A$ and $\mathbf{b} \in B$ are randomly chosen individuals. The distribution functions F and G of X and Y respectively are called the ***fitness distribution functions*** or simply ***fitness distributions.*** When they exist, the corresponding density functions f and g are called the ***fitness density functions*** or ***fitness densities.*** To each of these definitions corresponds an obvious "conditional" counterpart.

One type of conditional fitness distribution of interest is that for an individual with a given number of *ancestors* belonging to a particular class, since these are the fitness distributions of individuals surviving selection. Informally, an individual's ancestors are those individuals against which it competes, either explicitly or implicitly. This is made precise by the following definition.

Definition 5.1.3 (Ancestor): *Let* $\mathbf{a} \in P(t)$. *If* $t = 0$ *and* $\mathbf{a} \in A$ *then (in the population* $P(0)$*)* $\mathbf{a}$ *possesses one ancestor in* A *(itself). If* $t = 0$ *and* $\mathbf{a} \notin A$ *then (in the population* $P(0)$*)* $\mathbf{a}$ *possesses no ancestors in* A. *For* $t > 0$*:*

- *If in* $P(t-1)$ $\mathbf{a}$ *possesses* n *ancestors in* A *and it is selected in a tournament in which no probabilistically* θ*-compatible second individual is found then (in the population* $P(t)$*)* $\mathbf{a}$ *possesses* n *ancestors in* A *(the ancestors of* $\mathbf{a}$ *in* $P(t-1)$*).*
- *If in* $P(t-1)$ $\mathbf{a}$ *possesses* m *ancestors in* A, *where* $0 \leq m \leq n$, *and it is selected in a tournament in which the second individual* $\mathbf{b}$ *possesses* $n - m$ *ancestors in* A *then (in the population* $P(t)$*)* $\mathbf{a}$ *possesses* n *ancestors in* A *(the ancestors of* $\mathbf{a}$ *in* $P(t-1)$ *together with those of* $\mathbf{b}$ *in* $P(t-1)$*).*

□

Because the more fit individual wins each tournament with probability 1, every individual $\mathbf{a} \in P(t)$ is at least as fit as each of its ancestors. This straightforward observation leads to the following key element of the decision making model, which relates the conditional fitness distribution of an individual to the (unconditional) fitness distributions of its ancestors.

Corollary 5.1.4 *Let* **a** $\in A \subseteq I$, *and let* $B \subset I$ *such that* $A \cap B = \{\}$. *Then the conditional fitness distribution of* **a** *given that*

- **a** *has* $n_X \geq 1$ *ancestors in* A *(including itself), has* $n_Y \geq 0$ *ancestors in* B, *and has no other ancestors;*
- *those ancestors in* A *have fitness density* f *and fitness distribution* F;
- *those ancestors in* B *have fitness density* g *and fitness distribution* G; *and*
- *the fitnesses are mutually independent*

is

$$H(t) \quad = \quad K \int_{-\infty}^{t} f(x)[F(x)]^{n_X - 1}[G(x)]^{n_Y}\, dx \quad .$$

and the conditional density is

$$h(t) \quad = \quad K f(t)[F(t)]^{n_X - 1}[G(t)]^{n_Y} \quad ,$$

where

$$K \quad \triangleq \quad \frac{1}{\int_{-\infty}^{\infty} f(x)[F(x)]^{n_X - 1}[G(x)]^{n_Y}\, dx} \quad .$$

Proof: Let Z be the fitness of **a**, $(X_1, \ldots, X_{n_X - 1})$ the fitnesses of the remaining ancestors in A (if any), and $(Y_1, \ldots, Y_{n_Y})$ the fitnesses of those ancestors in B (if any). Because **a** is at least as fit as every ancestor, it is in particular at least as fit as any ancestors in A. Hence, $n_X > 0 \Longrightarrow Z \geq X_{n_X - 1 : n_X - 1}$. But **a** is also at least as fit as those ancestors in B, so $n_Y > 0 \Longrightarrow Z \geq Y_{n_Y : n_Y}$. The conclusion follows immediately from Theorem 5.1.2. ■

The next theorem states the probability of correct decision making as a function of the number of ancestors belonging to each class for each of the competing individuals. Without loss of generality, this research defines a "correct" decision to be one in which the individual in A is more fit than the one in B.

Theorem 5.1.5 *Let* $\mathbf{a} \in A \subset I$ *and* $\mathbf{b} \in B \subset I$, *where* $A \cap B = \{\}$. *Also, suppose that*

1. $\mathbf{a}$ *has* $n_X^{(\mathbf{a})} \geq 1$ *ancestors in* A, *has* $n_Y^{(\mathbf{a})} \geq 0$ *ancestors in* B, *and has no other ancestors;*
2. $\mathbf{b}$ *has* $n_X^{(\mathbf{b})} \geq 0$ *ancestors in* A, *has* $n_Y^{(\mathbf{b})} \geq 1$ *ancestors in* B, *and has no other ancestors;*
3. *those ancestors in* A *(of either* $\mathbf{a}$ *or* $\mathbf{b}$*) have fitness density* f *and fitness distribution* F*;*
4. *those ancestors in* B *(of either* $\mathbf{a}$ *or* $\mathbf{b}$*) have fitness density* g *and fitness distribution* G*; and*
5. *the fitnesses are mutually independent.*

Then the probability that $\mathbf{a}$ *is the more fit individual is*

$$p_d(n_X^{(\mathbf{a})}, n_Y^{(\mathbf{a})}, n_X^{(\mathbf{b})}, n_Y^{(\mathbf{b})}) = \\ K^{(\mathbf{a})} K^{(\mathbf{b})} \int_{-\infty}^{\infty} f(t)[F(t)]^{n_X^{(\mathbf{a})}-1}[G(t)]^{n_Y^{(\mathbf{a})}} \int_{-\infty}^{t} g(s)[F(s)]^{n_X^{(\mathbf{b})}}[G(s)]^{n_Y^{(\mathbf{b})}-1}\, ds\, dt \quad ,$$

where

$$K^{(\mathbf{a})} \triangleq \frac{1}{\int_{-\infty}^{\infty} f(x)[F(x)]^{n_X^{(\mathbf{a})}-1}[G(x)]^{n_Y^{(\mathbf{a})}}\, dx} \quad .$$

and

$$K^{(\mathbf{b})} \triangleq \frac{1}{\int_{-\infty}^{\infty} g(x)[F(x)]^{n_X^{(\mathbf{b})}}[G(x)]^{n_Y^{(\mathbf{b})}-1}\, dx} \quad .$$

Proof: Let h_A be the conditional fitness density of $\mathbf{a}$ given conditions 1, 3, 4, and 5. Also, let H_B be the conditional fitness distribution of $\mathbf{b}$ given conditions 2, 3, 4, and 5. Then the probability that $\mathbf{a}$ is more fit than $\mathbf{b}$ is just $\int_{-\infty}^{\infty} h_A(t)H_B(t)\,dt$. By Corollary 5.1.4,

$$h_A(t) = K^{(\mathbf{a})} f(t)[F(t)]^{n_X^{(\mathbf{a})}-1}[G(t)]^{n_Y^{(\mathbf{a})}}$$

and

$$H_B(t) = K^{(\mathbf{b})} \int_{-\infty}^{t} g(s)[G(s)]^{n_Y^{(\mathbf{b})}-1}[F(s)]^{n_X^{(\mathbf{b})}}\,ds \quad .$$

The result follows immediately. ■

5.1.3 Decision Making with Nontrivial Thresholding. This section extends the results of Section 5.1.2 to consider the effect of thresholding on the probability of correct decision making. Define the set $\Omega(D)$ of those pairs of individuals $(\mathbf{a},\mathbf{b})$ for which $\mathbf{a}$ belongs to A, $\mathbf{b}$ belongs to B, and the individual similarity is D. That is,

$$\Omega(D) \triangleq \{(\mathbf{a},\mathbf{b}) \in A \times B : d((\mathbf{a},\mathbf{b}) = D\} \quad ,$$

where d is the individual similarity, and it is understood that $A, B \subset I$ with $A \cap B = \{\}$.

Corollary 5.1.6 *Let* $\mathbf{a} \in A \subseteq I$, *and let* $B \subset I$ *such that* $A \cap B = \{\}$. *Then the conditional fitness distribution of* $\mathbf{a}$ *given that*

- $\mathbf{a}$ *has* $n_X \geq 1$ *ancestors in* A *(including itself), has* $n_Y \geq 0$ *ancestors in* B, *has no other ancestors, and has fitness density* f_Ω*;*
- *those ancestors in* A, *excluding* $\mathbf{a}$ *itself, have fitness density* f *and fitness distribution* F*;*
- *those ancestors in* B *have fitness density* g *and fitness distribution* G*; and*
- *the fitnesses are mutually independent*

is

$$H(t) \quad = \quad K \int_{-\infty}^{t} f_\Omega(x)[F(x)]^{n_X-1}[G(x)]^{n_Y}\,dx \quad .$$

and the conditional density is

$$h(t) \quad = \quad K f_\Omega(t)[F(t)]^{n_X-1}[G(t)]^{n_Y} \quad .$$

where

$$K \quad \triangleq \quad \frac{1}{\int_{-\infty}^{\infty} f_\Omega(x)[F(x)]^{n_X-1}[G(x)]^{n_Y}\, dx} \quad .$$

Proof: Let Z be the fitness of **a**, $(X_1, \ldots, X_{n_X-1})$ the fitnesses of the remaining ancestors in A (if any). and $(Y_1, \ldots, Y_{n_Y})$ the fitnesses of those ancestors in B (if any). Because **a** is at least as fit as every ancestor. it is in particular at least as fit as any ancestors in A. Hence, $n_X > 0 \Longrightarrow Z \geq X_{n_X-1:n_X-1}$. But **a** is also at least as fit as those ancestors in B, so $n_Y > 0 \Longrightarrow Z \geq Y_{n_Y:n_Y}$. The conclusion follows immediately from Theorem 5.1.2. ■

The next theorem states the probability of correct decision making as a function of the number of ancestors belonging to each class for each of the competing individuals and the similarity of the competing individuals.

Theorem 5.1.7 *Let* $(\mathbf{a}, \mathbf{b}) \in \Omega(D)$. *Also. suppose that*

1. **a** *has* $n_X^{(\mathbf{a})} \geq 1$ *ancestors in* A, *has* $n_Y^{(\mathbf{a})} \geq 0$ *ancestors in* B. *has no other ancestors, and has fitness density* f_Ω*:*

2. **b** *has* $n_X^{(\mathbf{b})} \geq 0$ *ancestors in* A, *has* $n_Y^{(\mathbf{b})} \geq 1$ *ancestors in* B. *has no other ancestors, and has fitness density* g_Ω*:*

3. *those ancestors in* A *(of either* **a** *or* **b**, *but excluding* **a** *itself) have fitness density* f *and fitness distribution* F*:*

4. *those ancestors in B (of either* **a** *or* **b**, *but excluding* **b** *itself) have fitness density g and fitness distribution G; and*

5. *the fitnesses are mutually independent.*

Then the probability that **a** *is the more fit individual is*

$$p_d(n_X^{(\mathbf{a})}, n_Y^{(\mathbf{a})}, n_X^{(\mathbf{b})}, n_Y^{(\mathbf{b})}) =$$
$$K^{(\mathbf{a})} K^{(\mathbf{b})} \int_{-\infty}^{\infty} f_\Omega(t)[F(t)]^{n_X^{(\mathbf{a})}-1}[G(t)]^{n_Y^{(\mathbf{a})}} \int_{-\infty}^{t} g_\Omega(s)[F(s)]^{n_X^{(\mathbf{b})}}[G(s)]^{n_Y^{(\mathbf{b})}-1}\, ds\, dt \quad .$$

where

$$K^{(\mathbf{a})} \triangleq \frac{1}{\int_{-\infty}^{\infty} f_\Omega(x)[F(x)]^{n_X^{(\mathbf{a})}-1}[G(x)]^{n_Y^{(\mathbf{a})}}\, dx} \quad ,$$

and

$$K^{(\mathbf{b})} \triangleq \frac{1}{\int_{-\infty}^{\infty} g_\Omega(x)[F(x)]^{n_X^{(\mathbf{b})}}[G(x)]^{n_Y^{(\mathbf{b})}-1}\, dx} \quad .$$

Proof: Let h_A be the conditional fitness density of **a** given conditions 1, 3, 4, and 5. Also, let H_B be the conditional fitness distribution of **b** given conditions 2, 3, 4, and 5. Then the probability that **a** is more fit than **b** is just $\int_{-\infty}^{\infty} h_A(t)H_B(t)\, dt$. By Corollary 5.1.4,

$$h_A(t) = K^{(\mathbf{a})} f_\Omega(t)[F(t)]^{n_X^{(\mathbf{a})}-1}[G(t)]^{n_Y^{(\mathbf{a})}}$$

and

$$H_B(t) = K^{(\mathbf{b})} \int_{-\infty}^{t} g_\Omega(s)[G(s)]^{n_Y^{(\mathbf{b})}-1}[F(s)]^{n_X^{(\mathbf{b})}}\, ds \quad .$$

The result follows immediately. ■

5.2 Dynamical Systems Model

In this section, the expected distribution of individuals between competing classes after selection is obtained. Specifically, the section investigates the effects of binary tournament selection (BTS) with probabilistic thresholding on the distribution of individuals belonging to classes A and B where $A \cap B = \{\}$ and $A \cup B = I$.

As in the analysis of building block filtering, the population is modeled via a *population vector* based on that proposed by Liepins and Vose [49]. The model is more general than that of Liepins and Vose in that the components of the population vector define a density function over a set of equivalence classes which form a partition of the augmented individual space $\hat{I} \triangleq I \times \mathbb{N} \times \mathbb{N}$.

Specifically, the population vector is of the form

$$\mathbf{p} \triangleq \left[\begin{array}{c} \mathbf{p}^{(\mathbf{A})} \\ \hline \mathbf{p}^{(\mathbf{B})} \end{array} \right] , \tag{12}$$

where

$$\mathbf{p}^{(\mathbf{A})} \triangleq \begin{bmatrix} \rho^{(A_{00})} & \cdots & \rho^{(A_{0j})} & \cdots \\ \vdots & \ddots & \vdots & \\ \rho^{(A_{i0})} & \cdots & \rho^{(A_{ij})} & \cdots \\ \vdots & & \vdots & \ddots \end{bmatrix}$$

and

$$\mathbf{p}^{(\mathbf{B})} \triangleq \begin{bmatrix} \rho^{(B_{00})} & \cdots & \rho^{(B_{0j})} & \cdots \\ \vdots & \ddots & \vdots & \\ \rho^{(B_{i0})} & \cdots & \rho^{(B_{ij})} & \cdots \\ \vdots & & \vdots & \ddots \end{bmatrix} .$$

Each component $\rho^{(A_{ij})}$ (resp. $\rho^{(B_{ij})}$ is the probability that an individual sampled from the population belongs to the equivalence class $A_{ij} \subset \hat{I}$ (resp. $B_{ij} \subset \hat{I}$) consisting of those individuals which are in A (resp. B), have i ancestors in A, and have j ancestors in B.

The effect of tournament selection on the population is modeled as a deterministic *transition function* τ_s mapping the current population vector p to the expected next population vector $\tau_s(p)$. By the law of large numbers, the next population vector in an infinite population algorithm exactly matches the expected population vector. Thus, the model developed here is exact for such algorithms.

Furthermore, as demonstrated by Vose and Wright [74], the transition function τ_s is independent of population size μ, assuming that the next population results from μ independent, identically distributed choices. For binary tournament selection, this assumption holds when the competing individuals are selected with replacement, regardless of population size. Hence, the model is also an exact expected value model for such algorithms.

Finally, for trivial thresholding (i.e. when all individuals are θ-compatible with probability 1), the expected population is independent of whether individuals are selected with or without replacement. Therefore, the model is also an exact expected value model for these algorithms.

The model is not necessarily exact for finite population size algorithms with nontrivial thresholding in which individuals are selected without replacement. Consider the case in which the most fit individual is θ-compatible with probability 1 with at least three other individuals, all other individuals are probabilistically θ-compatible with probability 0 with each other, and the shuffle size is very large. If individuals are selected with replacement, then the best individual is expected to compete in and win at least three tournaments. On the other hand, if individuals are selected without replacement, then the best individual is expected to compete in only two tournaments. Thus, the expected population under selection without replacement differs from the expected population under selection with replacement, which is predicted exactly by the proposed model.

In summary, the proposed model is exact (at least in expectation) for algorithms having any of the following properties:

- infinite population size,
- competing individuals are selected with replacement, or
- trivial thresholding.

Each tournament conducted by BTS with probabilistic thresholding may be viewed as a Markov chain, which this section analyzes at successively decreasing levels of abstraction (increasing levels of detail). The more abstract Markov chains (Section 5.2.1) view each tournament as a sequence of two state transitions — one corresponding to selection of the first individual, the other to the selection of the second individual and determination of the winner. The less abstract (more detailed) Markov chains (Section 5.2.2) focus on the state transitions required to search for a θ-compatible second individual before a winner can be determined.

5.2.1 High-level Markov Chain.

The most abstract view of a binary tournament (of those considered here), which is called MC–0, is represented as a state transition diagram in Figure 20. The five states of MC–0 are:

- s_0: Initial state.
- $s^{(A)}$: The first individual is in A.[2]
- $s^{(B)}$: The first individual is in B.
- s_A: The winner is in A.
- s_B: The winner is in B.

The transition probability from s_0 to $s^{(A)}$ is the probability $\rho^{(A)}(t) \triangleq \sum_{i,j} \rho^{(A_{ij})}(t)$ that an individual drawn from the current population $P(t)$ is in A. For the initial iteration of selection ($t = 0$), this probability

[2] Throughout this section, single superscripts refer to the first individual. The first (second) element of an ordered pair superscript refers to the first (second) individual. Finally, subscripts refer to the tournament winner.

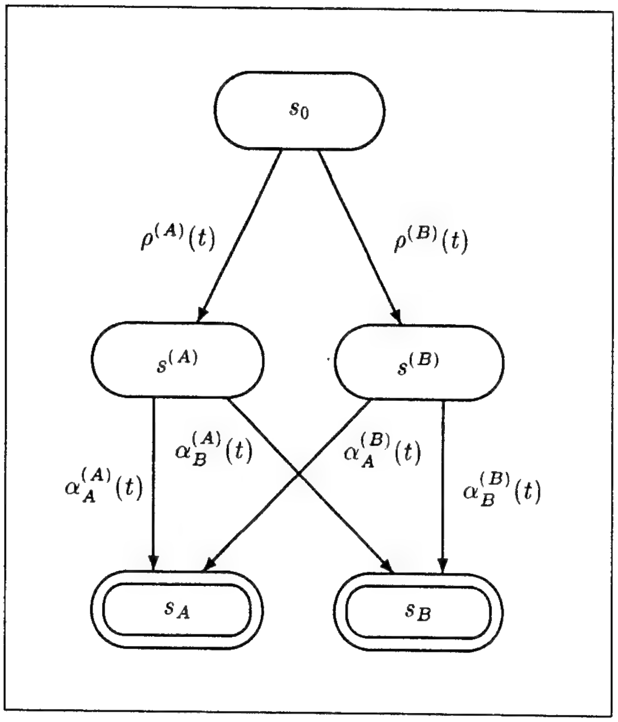


Figure 20. MC–0: High-level Markov Chain Model of a Binary Tournament.

is determined entirely by the distribution from which the initial population $P(0)$ is drawn (Section 5.3). Thereafter, it is determined by the following recurrence relation.

Theorem 5.2.1 *Let $\alpha_A^{(A)}(t)$ be the conditional probability that the winner of a tournament in generation t is in A given that the first individual is in A. Also, let $\alpha_A^{(B)}(t)$ be the conditional probability that the winner of a tournament in generation t is in A given that the first individual is in B. Finally, let $\mathbf{a}$ be randomly drawn from the population $P(t+1)$, where $t \geq 0$. Then the probability that $\mathbf{a} \in A$ is*

$$\rho^{(A)}(t+1) \quad = \quad \rho^{(A)}(t)\left[\alpha_A^{(A)}(t) - \alpha_A^{(B)}(t)\right] + \alpha_A^{(B)}(t) \quad .$$

Proof: The probability that an individual drawn from population $P(t+1)$ is in A is just the probability that the winner of a tournament in generation t is in A. The latter is the absorption probability from s_0

into s_A in generation t, so that

$$\begin{aligned}
\rho^{(A)}(t+1) &= \rho^{(A)}(t)\alpha_A^{(A)}(t) + \rho^{(B)}(t)\alpha_A^{(B)}(t) \\
&= \rho^{(A)}(t)\alpha_A^{(A)}(t) + \left[1 - \rho^{(A)}(t)\right]\alpha_A^{(B)}(t) \\
&= \rho^{(A)}(t)\left[\alpha_A^{(A)}(t) - \alpha_A^{(B)}(t)\right] + \alpha_A^{(B)}(t) \quad .
\end{aligned}$$

■

Due to symmetry, an analogous relation holds for $\rho^{(B)}$.

The transition probability $\alpha_A^{(A)}(t)$ is the conditional probability that the winner of a tournament in generation t is in A given that the first individual is in A. and $\alpha_A^{(B)}(t)$ is the corresponding probability given that the first individual is in B. Both depend on the probabilities of correct decision making, and thus on the number n_A of the first individual's ancestors in A and the number n_B in B (see Section 5.1). Thus, it is necessary to consider the less abstract (more refined) Markov chain MC–1 which explicitly depicts these dependencies (see Figure 21). The states of MC–1 are:

- s_0: Initial state.
- $s^{(A_{n_A n_B})}, n_A \geq 1, n_B \geq 0. n_A + n_B \leq 2^t$: The first individual is in A. has n_A ancestors in A. and has n_B ancestors in B.
- $s^{(B_{n_A n_B})}. n_A \geq 0, n_B \geq 1, n_A + n_B \leq 2^t$: The first individual is in B. has n_A ancestors in A, and has n_B ancestors in B.
- $s_{A_{ij}}, i \geq 1, j \geq 0, i + j \leq 2^{t+1}$: The winner is in A, has i ancestors in A, and has j ancestors in B.
- $s_{B_{ij}}. i \geq 0, j \geq 1, i + j \leq 2^{t+1}$: The winner is in B, has i ancestors in A. and has j ancestors in B.

As previously mentioned, the transition probability $\rho^{(A_{n_A n_B})}(t)$ from s_0 to $s^{(A_{n_A n_B})}$ is the probability that an individual drawn from the population $P(t)$ is in A, has n_A ancestors in A. and has n_B ancestors in B.

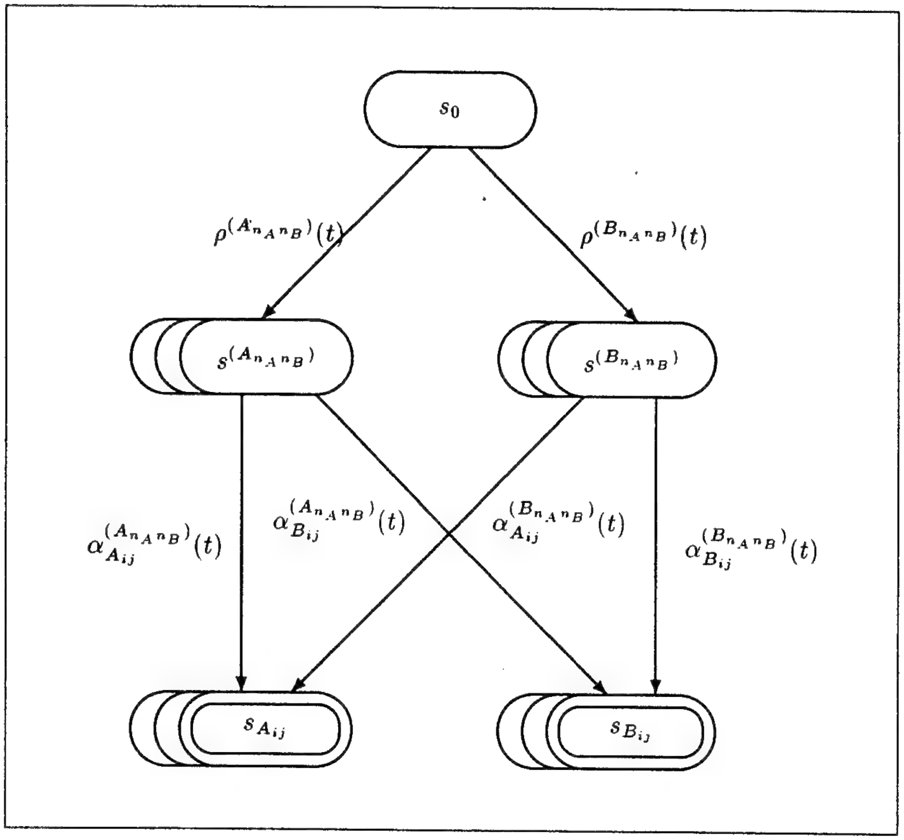

s_0
$\rho^{(A_{n_A n_B})}(t)$
$\rho^{(B_{n_A n_B})}(t)$
$s^{(A_{n_A n_B})}$
$s^{(B_{n_A n_B})}$
$\alpha_{A_{ij}}^{(A_{n_A n_B})}(t)$
$\alpha_{B_{ij}}^{(A_{n_A n_B})}(t)$
$\alpha_{A_{ij}}^{(B_{n_A n_B})}(t)$
$\alpha_{B_{ij}}^{(B_{n_A n_B})}(t)$
$s_{A_{ij}}$
$s_{B_{ij}}$

Figure 21. MC–1: High-level Markov chain model of a binary tournament explicitly depicting the dependence of the transition probabilities on the first individual's ancestors.

Like $\rho^{(A)}(t)$. it satisfies a recurrence relation, given in the following theorem. Again, the initial condition is determined by the distribution from which the initial population $P(0)$ is drawn.

Theorem 5.2.2 *Let $\rho^{(A)}(0)$ be the probability that an individual drawn from the initial population $P(0)$ is in A. Also. let* **a** *be drawn randomly from $P(0)$. Then the probability that* **a** $\in A_{ij}$ *is*

$$\rho^{(A_{ij})}(0) \quad = \quad \begin{cases} \rho^{(A)}(0) & \text{, } \textit{if } i = 1 \textit{ and } j = 0 \\ 0 & \text{, } \textit{otherwise} \end{cases} .$$

Furthermore. let $\alpha_{A_{ij}}^{(A_{n_A n_B})}(t)$ be the conditional probability that the winner of a tournament in generation t is in A_{ij} given that the first individual is in $A_{n_A n_B}$, and let $\alpha_{A_{ij}}^{(B_{n_A n_B})}(t)$ be the conditional probability that *the winner of a tournament in generation t is in A_{ij} given that the first individual is in $B_{n_A n_B}$ Finally, let* $\mathbf{a}$ *be drawn randomly from $P(t+1)$, where $t \geq 0$. Then the probability that* $\mathbf{a} \in A_{ij}$ *is*

$$\rho^{(A_{ij})}(t+1) = \sum_{n_A}\sum_{n_B}\left[\rho^{(A_{n_A n_B})}(t)\alpha_{A_{ij}}^{(A_{n_A n_B})}(t) + \rho^{(B_{n_A n_B})}(t)\alpha_{A_{ij}}^{(B_{n_A n_B})}(t)\right] .$$

Proof: Every individual in the initial population $P(0)$ has exactly one ancestor (itself). Also. the probability that an individual drawn from population $P(t+1)$ is in A_{ij} is just the probability that the winner of a tournament in generation t is in A_{ij}. The latter is the absorption probability from s_0 into $s_{A_{ij}}$ in generation t. ■

Again. due to symmetry, an analogous result holds for the $\rho^{(B_{ij})}(t)$'s.

5.2.2 Low-level Markov Chain. The transition from $s^{(A_{n_A n_B})}$ to $s_{A_{ij}}$ or $s_{B_{ij}}$ in MC–1 involves the intermediate steps required to search for a θ-compatible second individual. These steps form a Markov chain MC–2 for which some of the transition probabilities depend on n_A and n_B. Figure 22 shows MC–2, which may be viewed as a fragment of a still more refined Markov chain model of the overall tournament. The states of MC–2 are:

- $s^{(A_{n_A, n_B})}$: Initial state. The first individual is in $A_{n_A n_B}$. No candidate second individuals have been considered.
- $s_r^{(A_{n_A, n_B})}, 0 \leq r \leq n_{sh}$: The first individual is in $A_{n_A n_B}$. Furthermore. r second individuals have been considered and probabilistically found not to be θ-compatible.
- $s_{A_{ij}}$(and $s_{A_{n_A n_B}}$),$i \geq 1, j \geq 0, i+j \leq 2^t$: The winner is in A_{ij} (resp. $A_{n_A n_B}$).
- $s_{B_{ij}}, i \geq 0, j \geq 1, i+j \leq 2^t$: The winner is in B_{ij}.

The transition probability $\alpha_{A_{ij}}^{(A_{n_A n_B})}(t)$ in MC–1 is the conditional probability that the winner of a tournament in generation t is in A_{ij} given that the first individual is in $A_{n_A n_B}$. The following theorem provides an expression for $\alpha_{A_{ij}}^{(A_{n_A n_B})}(t)$ in terms of the transition probabilities of MC–2.

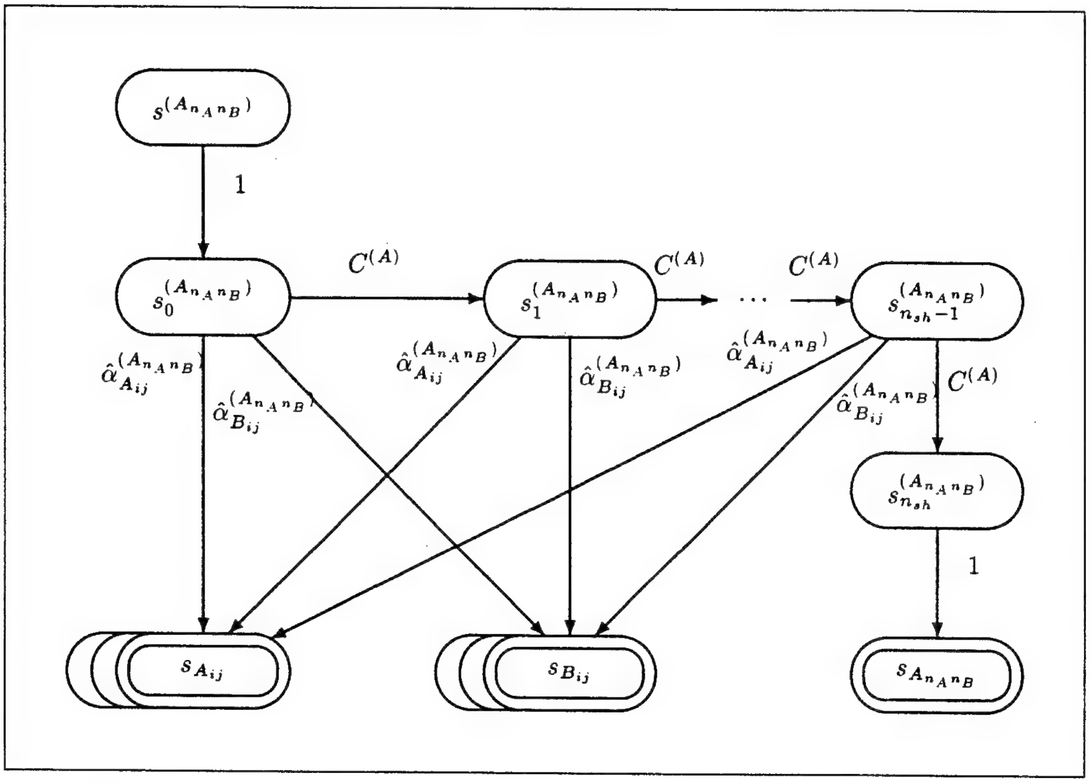

$s^{(A_{n_A n_B})}$
1
$s_0^{(A_{n_A n_B})}$
$C^{(A)}$
$s_1^{(A_{n_A n_B})}$
$C^{(A)}$
$\ldots$
$C^{(A)}$
$s_{n_{sh}-1}^{(A_{n_A n_B})}$
$\hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}$
$\hat{\alpha}_{B_{ij}}^{(A_{n_A n_B})}$
$\hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}$
$\hat{\alpha}_{B_{ij}}^{(A_{n_A n_B})}$
$\hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}$
$\hat{\alpha}_{B_{ij}}^{(A_{n_A n_B})}$
$C^{(A)}$
$s_{n_{sh}}^{(A_{n_A n_B})}$
1
$s_{A_{ij}}$
$s_{B_{ij}}$
$s_{A_{n_A n_B}}$

Figure 22. MC–2: Markov chain model of the search for a probabilistically θ-compatible second individual and the determination of the tournament winner. The first individual is in A, has n_A ancestors which are in A, and has n_B ancestors in B. Dependence of the transition probabilities on the iteration t is notationally suppressed for visual clarity.

Theorem 5.2.3 *Suppose that*

- $\hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}(t)$ *is the conditional probability that a candidate second individual in generation* t *is probabilistically* θ*-compatible and the tournament winner is in* A_{ij} *given that the first individual is in* $A_{n_A n_B}$.
- $C^{(A)}$ *is the conditional probability that a candidate second individual in generation* t *is not probabilistically* θ*-compatible with the first individual given that the first individual is in* $A_{n_A n_B}$, *and*
- n_{sh} *is the shuffle size.*

Then the conditional probability that the winner of a tournament in generation t is in A_{ij} given that the first individual is in $A_{n_A n_B}$ is

$$\alpha_{A_{ij}}^{(A_{n_A n_B})}(t) = \hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}(t) \sum_{r=0}^{n_{sh}-1} \left[C^{(A)}\right]^r + \delta_{in_A}\delta_{jn_B}\left[C^{(A)}\right]^{n_{sh}} ,$$

where δ_{ij} is the Kronecker delta.

Proof: The transition from state $s^{(A_{n_A n_B})}$ to state $s_{A_{ij}}$ in MC–1 (see Figure 21) is equivalent to the corresponding absorption event in MC–2 (see Figure 22). Thus, the transition probability $\alpha_{A_{ij}}^{(A_{n_A n_B})}(t)$ is equal to the probability of the absorption event, i.e.

$$\begin{aligned}\alpha_{A_{ij}}^{(A_{n_A n_B})}(t) &= 1\left[\sum_{r=0}^{n_{sh}-1} \left[C^{(A)}\right]^r \hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}(t) + \delta_{in_A}\delta_{jn_B}\left[C^{(A)}\right]^{n_{sh}} \cdot 1\right] \\ &= \hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}(t) \sum_{r=0}^{n_{sh}-1} \left[C^{(A)}\right]^r + \delta_{in_A}\delta_{jn_B}\left[C^{(A)}\right]^{n_{sh}} .\end{aligned}$$

■

The next theorem provides a similar result for the MC–1 transition probability $\alpha_{B_{ij}}^{(A_{n_A n_B})}(t)$, which is the conditional probability that the winner of a tournament in generation t is in B_{ij}, again given that the first individual is in $A_{n_A n_B}$.

Theorem 5.2.4 *Suppose that*

- $\hat{\alpha}_{B_{ij}}^{(A_{n_A n_B})}(t)$ *is the conditional probability that a candidate second individual in generation t is probabilistically θ-compatible and the tournament winner is in B_{ij} given that the first individual is in $A_{n_A n_B}$, and*

- $C^{(A)}$ *is the conditional probability that a candidate second individual is not probabilistically θ-compatible given that the first individual is in $A_{n_A n_B}$.*

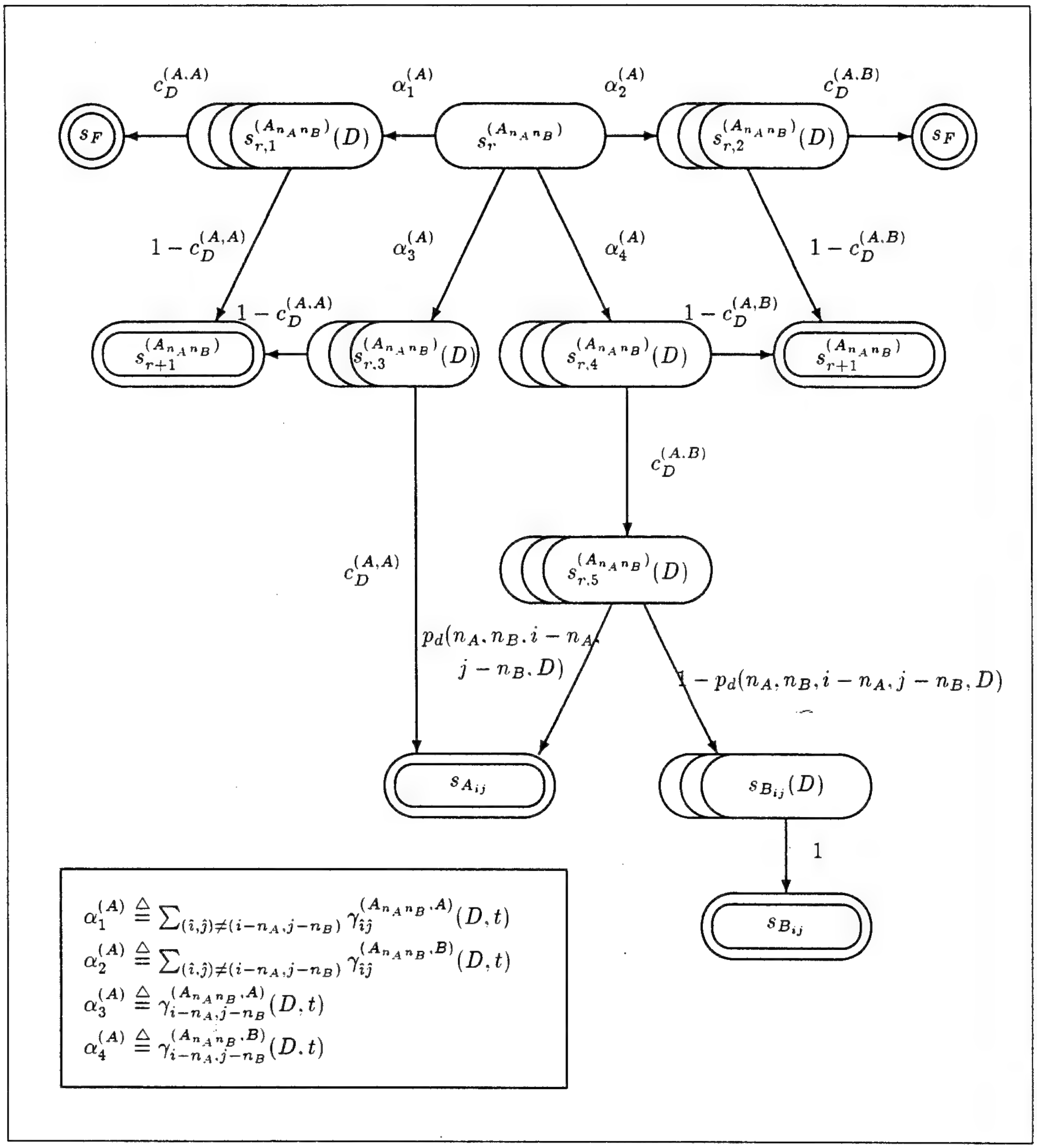
Then the conditional probability that the winner of a tournament in generation t is in B_{ij} given that the first individual is in $A_{n_A n_B}$ is

$$\alpha_{B_{ij}}^{(A_{n_A n_B})}(t) \quad = \quad \hat{\alpha}_{B_{ij}}^{(A_{n_A n_B})}(t) \sum_{r=0}^{n_{sh}-1} \left[C^{(A)}\right]^r \quad .$$

For each $r \in \{0, \ldots, n_{sh} - 1\}$, the transition from $s_r^{(A_{n_A n_B})}$ to $s_{A_{ij}}$, $s_{B_{ij}}$, or $s_{r+1}^{(A_{n_A n_B})}$ involves sampling the population to choose a candidate second individual. determination of compatibility, and (possibly) determination of the more fit individual. Furthermore, the probability of correct decision making depends on the thresholding distance D (see Section 5.1).

Thus, it is necessary to consider another "lower-level" (more refined) Markov chain MC–3 in which these steps and dependencies appear explicitly. For $r < n_{sh}$ the transition probabilities do not depend on r. Thus it is sufficient to consider a representative fragment for which the first individual is in $A_{n_A n_B}$ and r candidate second individuals have been considered and found not to be probabilistically θ-compatible. MC–3 is shown in Figure 23. Its states are:

- $s_r^{(A_{n_A n_B})}$: Initial state. The first individual is in $A_{n_A n_B}$, and r candidate second individuals have been considered and found not to be probabilistically θ-compatible.



$c_D^{(A,A)}$
$\alpha_1^{(A)}$
$\alpha_2^{(A)}$
$c_D^{(A,B)}$
s_F
$s_{r,1}^{(A_{n_A n_B})}(D)$
$s_r^{(A_{n_A n_B})}$
$s_{r,2}^{(A_{n_A n_B})}(D)$
s_F
$1 - c_D^{(A,A)}$
$\alpha_3^{(A)}$
$\alpha_4^{(A)}$
$1 - c_D^{(A,B)}$
$1 - c_D^{(A,A)}$
$1 - c_D^{(A,B)}$
$s_{r+1}^{(A_{n_A n_B})}$
$s_{r,3}^{(A_{n_A n_B})}(D)$
$s_{r,4}^{(A_{n_A n_B})}(D)$
$s_{r+1}^{(A_{n_A n_B})}$
$c_D^{(A,B)}$
$c_D^{(A,A)}$
$s_{r,5}^{(A_{n_A n_B})}(D)$
$p_d(n_A, n_B, i - n_A, j - n_B, D)$
$1 - p_d(n_A, n_B, i - n_A, j - n_B, D)$
$s_{A_{ij}}$
$s_{B_{ij}}(D)$
1
$s_{B_{ij}}$
$\alpha_1^{(A)} \triangleq \sum_{(i,j)\neq(i-n_A,j-n_B)} \gamma_{ij}^{(A_{n_A n_B},A)}(D,t)$
$\alpha_2^{(A)} \triangleq \sum_{(i,j)\neq(i-n_A,j-n_B)} \gamma_{ij}^{(A_{n_A n_B},B)}(D,t)$
$\alpha_3^{(A)} \triangleq \gamma_{i-n_A,j-n_B}^{(A_{n_A n_B},A)}(D,t)$
$\alpha_4^{(A)} \triangleq \gamma_{i-n_A,j-n_B}^{(A_{n_A n_B},B)}(D,t)$

Figure 23. MC–3: Fragment of "low-level" Markov chain model of selection of the second individual in BTS with thresholding and finite shuffle size. The first individual is in A, has n_A ancestors in A, and has n_B ancestors in B. Also, r candidate second individuals have been found not to be probabilistically θ-compatible.

- $s_{r,1}^{(A_{n_A n_B})}(D).0 \leq D \leq D_{\max}$: The conditions of $s_r^{(A_{n_A n_B})}$ hold. Furthermore, the candidate second individual has individual similarity D from the first individual and is in $A - A_{i-n_A,j-n_B}$ (i.e. is in A and either 1) has other than $i - n_A$ ancestors in A, or 2) has other than $j - n_B$ ancestors in B).

- $s_{r,2}^{(A_{n_A n_B})}(D).0 \leq D \leq D_{\max}$: The conditions of $s_r^{(A_{n_A n_B})}$ hold. Furthermore, the candidate second individual has individual similarity D from the first individual and is in $B - B_{i-n_A,j-n_B}$.

- $s_{r,3}^{(A_{n_A n_B})}(D).0 \leq D \leq D_{\max}$: The conditions of $s_r^{(A_{n_A n_B})}$ hold. Furthermore, the candidate second individual has individual similarity D from the first individual and is in $A_{i-n_A,j-n_B}$.

- $s_{r,4}^{(A_{n_A n_B})}(D).0 \leq D \leq D_{\max}$: The conditions of $s_r^{(A_{n_A n_B})}$ hold. Furthermore, the candidate second individual has individual similarity D from the first individual and is in $B_{i-n_A,j-n_B}$.

- $s_{r,5}^{(A_{n_A n_B})}(D).0 \leq D \leq D_{\max}$: The conditions of state $s_{r,4}^{(A_{n_A \cdot n_B})}(D)$ hold. Furthermore, the individuals are probabilistically θ-compatible, hence a "decision" must be made.

- $s_{r+1}^{(A_{n_A n_B})}$ (depicted twice): The candidate second individual is not probabilistically θ-compatible. This state coincides with the "next" state of MC–2 (Figure 22).

- $s_{A_{ij}}$: The tournament winner is in A_{ij}.

- $s_{B_{ij}}$: The tournament winner is in B_{ij}.

- s_F (depicted twice): Represents the union of all MC–2 final states not explicitly depicted. i.e.

$$S_F \triangleq \bigcup_{(\hat{i},\hat{j}) \neq (i,j)} \{s_{A_{\hat{i}\hat{j}}}, s_{B_{\hat{i}\hat{j}}}\} \ .$$

The transition probability $\hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}(t)$ in MC–2 is the conditional probability that a candidate second individual in generation t is probabilistically θ-compatible and the tournament winner is in B_{ij} given that the first individual is in $A_{n_A n_B}$. The following theorem provides an expression for $\alpha_{A_{ij}}^{(A_{n_A n_B})}(t)$ in terms of the transition probabilities of MC–3.

Theorem 5.2.5 *Suppose that*

- $\gamma^{(A_{n_A n_B}),A}_{i-n_A,j-n_B}(D,t)$ *is the conditional probability that an individual drawn from the current population has individual similarity D from the first individual and is in $A_{i-n_A,j-n_B}$, given the conditions of state $s_r^{(A_{n_A n_B})}$;*
- $\gamma^{(A_{n_A n_B}),B}_{i-n_A,j-n_B}(D,t)$ *is the conditional probability that an individual drawn from the current population has individual similarity D from the first individual and is in $B_{i-n_A,j-n_B}$, given the conditions of state $s_r^{(A_{n_A n_B})}$;*
- $c_D^{(A,A)}$ *is the conditional probability that the individuals are probabilistically θ-compatible given given the conditions of state $s_{r,3}^{(A_{n_A n_B})}(D)$;*
- $c_D^{(A,B)}$ *is the conditional probability that the individuals are probabilistically θ-compatible given the conditions of state $s_{r,4}^{(A_{n_A n_B})}(D)$; and*
- $p_d(n_A,n_B,i-n_A,j-n_B,D)$ *is the conditional probability that the first individual is more fit (see Section 5.1) given the conditions of state $s_{r,5}^{A_{n_A n_B}}(D)$.*

Then the conditional probability in generation t that a candidate second individual is probabilistically θ-compatible and the tournament winner is in A_{ij} given that the first individual in is $A_{n_A n_B}$ is

$$\hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}(t) = \sum_D \left[\gamma^{(A_{n_A n_B},A)}_{i-n_A,j-n_B}(D,t)c_D^{(A,A)} + \gamma^{(A_{n_A n_B},B)}_{i-n_A,j-n_B}(D,t)c_D^{(A,B)}p_d(n_A,n_B,i-n_A,j-n_B,D)\right] .$$

Proof: The transition event from $s_r^{(A_{n_A},n_B)}$ to state $s_{A_{ij}}$ in MC–2 (see Figure 22) is equivalent to the corresponding absorption event in MC–3 (see Figure 23). Thus, the transition probability $\hat{\alpha}_{A_{ij}}^{(A_{n_A n_B})}(t)$ is equal to the probability of the absorption event. ■

The next theorem provides a similar result for the MC–2 transition probability $\hat{\alpha}_{B_{ij}}^{(A_{n_A n_B})}(t)$, which is the conditional probability that a candidate second individual in generation t is probabilistically θ-compatible and the tournament winner is in B_{ij}, again given that the first individual is in $A_{n_A n_B}$.

Theorem 5.2.6 *Suppose that*

- $\gamma_{i-n_A,j-n_B}^{(A_{n_An_B}),B}(D,t)$ *is the conditional probability that an individual drawn from the current population has individual similarity* D *from the first individual and is in* $B_{i-n_A,j-n_B}$, *given that the first individual is in* $A_{n_An_B}$, *and* r *candidate second individuals have been considered and found not to be probabilistically* θ*-compatible;*
- $c_D^{(A,B)}$ *is the conditional probability that the individuals are probabilistically* θ*-compatible given the conditions of state* $s_{r,4}^{(A_{n_An_B})}(D)$*; and*
- $p_d(n_A,n_B,i-n_A,j-n_B,D)$ *is the conditional probability that the first individual is more fit (see Section 5.1) given the conditions of state* $s_{r,5}^{A_{n_An_B}}(D)$.

Then the conditional probability in generation t *that a candidate second individual is probabilistically* θ*-compatible and the tournament winner is in* B_{ij} *given that the first individual in is* $A_{n_An_B}$ *is*

$$\hat{\alpha}_{B_{ij}}^{(A_{n_An_B})}(t) = \sum_D \gamma_{i-n_A,j-n_B}^{(A_{n_An_B},B)}(D,t)c_D^{(A,B)}[1-p_d(n_A,n_B,i-n_A,j-n_B,D)] \;.$$

Proof: The transition event from $s_r^{(A_{n_A,n_B})}$ to state $s_{B_{ij}}$ in MC–2 (see Figure 22) is equivalent to the corresponding absorption event in MC–3 (see Figure 23). ■

Finally, the following theorem provides the probability that a candidate second individual is not probabilistically θ-compatible.

Theorem 5.2.7 *The conditional probability that a candidate second individual is not probabilistically* θ*-compatible given that the first individual is in* $A_{n_An_B}$ *is*

$$C^{(A)} = 1 - \sum_D \left[c_D^{(A,A)} \sum_i \sum_j \gamma_{ij}^{(A_{n_An_B},A)}(D,t) + c_D^{(A,B)} \sum_i \sum_j \gamma_{i-n_A,j-n_B}^{(A_{n_An_B})}(D,t) \right] \;.$$

Proof: The conditional probability that a candidate second individual is not probabilistically θ-compatible, given that the first individual is in $A_{n_A n_B}$, is equal to the transition probability from $s_{r,0}^{(A)}(n_A, n_B)$ to $s_{r+1,0}^{(A)}(n_A, n_B)$. This is given by

$$\begin{aligned}
C^{(A)} &= \sum_D \sum_i \sum_j \left\{ \gamma_{ij}^{(A_{n_A n_B},A)}(D,t) \left[1 - c_D^{(A,A)}\right] + \gamma_{ij}^{(A_{n_A n_B},B)}(D,t) \left[1 - c_D^{(A,B)}\right] \right\} \\
&= 1 \cdot \sum_D \sum_i \sum_j \left[\gamma_{ij}^{(A_{n_A n_B},A)}(D,t) + \gamma_{ij}^{(A_{n_A n_B})}(D,t) \right] \\
&\quad - \sum_D \left[c_D^{(A,A)} \sum_i \sum_j \gamma_{ij}^{(A_{n_A n_B},A)}(D,t) + c_D^{(A,B)} \sum_i \sum_j \gamma_{i-n_A,j-n_B}^{(A_{n_A n_B})}(D,t) \right] \\
&= 1 - \sum_D \left[c_D^{(A,A)} \sum_i \sum_j \gamma_{ij}^{(A_{n_A n_B},A)}(D,t) + c_D^{(A,B)} \sum_i \sum_j \gamma_{i-n_A,j-n_B}^{(A_{n_A n_B})}(D,t) \right] ,
\end{aligned}$$

which does not depend on r. ■

This section concludes with the observation that Kargupta's model of BTS (Section 2.6.5.2) may be obtained as a special case of the model developed in this chapter by assuming that

- a compatible second individual is found for every tournament,
- no individual contains multiple building blocks,
- every individual which contains any building block is more fit than every individual which contains no building blocks, and
- the probabilities of correct decision making are static (i.e. independent of the generation t).

5.3 Distribution of Fitnesses in a Uniform Random Population

The probability of correct decision making, and consequently the distribution of individuals between competing classes, depends on the fitness distributions of the ancestors of the competing individuals (see Sections 5.1 and 5.2). The fitness distributions are determined by the distribution from which the initial population $P(0)$ is drawn, as well as the fitness function itself.

This section presents the fitness distributions obtained in the case of linkage-friendly genetic algorithms. The analysis focuses on the first two central moments (i.e. the expected values and variances) of the fitness distributions resulting when then fitness function is separable (see Section 5.3.1). The competing classes under consideration are the class I_β of individuals containing building block β and the class $I_{\neg\beta}$ of individuals lacking building block β. These classes are formally defined in Section 5.3.1, along with several other classes which appear frequently in the analysis. In each case, two distributions are considered. The first is the (unconditional) fitness distribution of the class (Section 5.3.2). The other is the conditional fitness distribution given also that the individual is a member of a pair of individuals drawn randomly from the set of pairs having individual similarity D (Section 5.3.3).

5.3.1 Preliminaries. As discussed in Section 2.6, the class of linkage-friendly genetic algorithms includes the fast messy genetic algorithm (Section 2.6.4). More generally, it includes the generalized fast messy genetic algorithm proposed in Chapter III. Three properties of these algorithms which are of use in the analysis of this section are:

- the individual space I consists of ordered pairs of finite sequences. where elements of one sequence are alleles, and elements of the other sequence are loci (see Section 2.6.1);
- evaluation of individuals in which one or more loci do not occur relies on default values specified by the competitive template (see Section 2.6.1): and
- the initial population is uniformly distributed over the set $I(\lambda)$ of length λ non-overspecified individuals.

The notations $I(\lambda)$, I_F, $I_\beta(\lambda)$, and $I_{\neg\beta}(\lambda)$ for important subsets of the individual space I are introduced in Sections 2.6.1 and 4.1.1. It is convenient to introduce special notation for other frequently mentioned subsets of I, which simplifies the analysis presented in the sequel.

- For each $\lambda \in \{0, \ldots, \ell\}$ and each $i \in \{1, \ldots, m\}$, let $\mathbf{c} \in I_F$ and define

$$I_{\chi_i}(\lambda, \mathbf{c}) \triangleq \{(\mathbf{a}, \mathbf{l}) \in I(\lambda) : \neg(\forall L \in \mathcal{L}_i)(\forall j \in \mathcal{L})[l_j = L \Longrightarrow a_j = c_L]\} \quad .$$

Then $I_{\chi_i}(\lambda.\mathbf{c})$ is the set of length λ individuals which *disrupt the competitive template* $\mathbf{c}$ *with respect to subfunction i.*

- For each $\lambda \in \{0.\ldots.\ell\}$ and each $i \in \{1,\ldots,m\}$, define

$$I_{\neg\chi_i}(\lambda,\mathbf{c}) \triangleq I(\lambda) - I_{\chi_i}(\lambda,\mathbf{c}) \quad .$$

  Then $I_{\neg\chi_i}(\lambda,\mathbf{c})$ is the set of length λ individuals which do not disrupt the competitive template $\mathbf{c}$ with respect to subfunction i.

Certain intersections of these sets are also frequently mentioned. These intersections are denoted by concatenation of subscripts. When subscripts refer to (possibly) different subfunctions, a comma is inserted for clarity, e.g. for each $\lambda \in \{0.\ldots.\ell\}$ and each $i \in \{1,\ldots,m\}$,

$$I_{\beta,\neg i\chi_i}(\lambda,\mathbf{c}) \triangleq I_\beta(\lambda) \cap I_{\neg i}(\lambda) \cap I_{\chi_i}(\lambda.\mathbf{c})$$

is the set of length λ individuals which contain building block β. lack building block i. and disrupt the competitive template $\mathbf{c}$ with respect to subfunction i.

It is also convenient to introduce special notation for the cardinalities[3] of these sets. In general, the number of individuals contained in a set $I_S(\lambda)$ is denoted $N_S(\lambda)$, e.g. the number of length λ individuals containing building block β. lacking building block i. and disrupting the competitive template $\mathbf{c}$ with respect to subfunction i is $N_{\beta,\neg i\chi_i}(\lambda.\mathbf{c})$. Analytical expressions for those cardinalities which appear in the analysis are given in Appendix A.

Also, the set of pairs of individuals which share some number of defining loci appears frequently in the sequel. Let $\Lambda_c(\mathbf{x_1}.\mathbf{x_2})$ denote the number of *common defining loci* of individuals $\mathbf{x_1}$ and $\mathbf{x_2}$. That is.

$$\Lambda_c((\mathbf{a_1}.\mathbf{l_1}).(\mathbf{a_2},\mathbf{l_2})) \quad \triangleq \quad \mathrm{card}\left(\{L \in \mathcal{L} : (\exists i_1.i_2 \in \mathcal{L})[l_{1,i_1} = l_{2.i_2} = L]\}\right) \quad .$$

[3] This research considers only the case of a finite genic alphabet $\mathcal{A}$.

Also, let $\omega(\lambda_1, \lambda_2, \lambda_c)$ be the set of individual pairs $(\mathbf{x_1}, \mathbf{x_2})$ for which $\mathbf{x_1}$ contains building block β and is of length λ_1, $\mathbf{x_2}$ does not contain building block β and is of length λ_2, and $\Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c$. That is,

$$\omega(\lambda_1, \lambda_2, \lambda_c) \triangleq \{(\mathbf{x_1}, \mathbf{x_2}) \in I_\beta(\lambda_1) \times I_{\neg\beta}(\lambda_2) : \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c\} \ .$$

The set $\omega(\lambda_1, \lambda_2, \lambda_c)$ is related to the set $\Omega(D)$ consisting of pairs of individuals in $A \times B$ having individual similarity D (see Section 5.1). In particular, if $A = I_\beta(\lambda_1)$, $B = I_{\neg\beta}(\lambda_2)$, and $D = d(\mathbf{x_1}, \mathbf{x_2}) \triangleq \min\{\lambda_1, \lambda_2\} - \Lambda_c(\mathbf{x_1}, \mathbf{x_2})$, then $\Omega(D) = \omega(\lambda_1, \lambda_2, \lambda_c)$.

Finally, the fitness distributions considered in Sections 5.3.2 and 5.3.3 are those associated with fitness functions which can be written as the sum of independent subfunctions. The following results are used so frequently in their analysis that they are stated as lemmas. The first shows that certain interesting conditional variances vanish.

Lemma 5.3.1 (Special conditional variances of subfunction contributions) *Let* $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ *be a separable fitness function,* $\mathbf{c} \in I_F$, *and* $\mathbf{X} \sim U(I(\lambda))$. *Then the conditional variance of* $\phi_i(\mathbf{X}, \mathbf{c})$ *given that* $\mathbf{X}$ *contains building block* i *is zero. Also, the conditional variance of* $\phi_i(\mathbf{X}, \mathbf{c})$ *given that* $\mathbf{X}$ *does not disrupt* $\mathbf{c}$ *with respect to subfunction* i *is zero. Finally, the conditional variance of* $\phi_i(\mathbf{X}, \mathbf{c})$ *given that* $\mathbf{X}$ *lacks building block* i *and does not disrupt* $\mathbf{c}$ *with respect to subfunction* i *is also zero, i.e.*

$$Var\,[\phi_i(\mathbf{X}, \mathbf{c}) \mid \mathbf{X} \in I_i(\lambda)] = Var\,[\phi_i(\mathbf{X}, \mathbf{c}) \mid \mathbf{X} \in I_{\neg\chi_i}(\lambda)] = Var\,[\phi_i(\mathbf{X}, \mathbf{c}) \mid \mathbf{X} \in I_{\neg i \neg\chi_i}(\lambda)] = 0 \ .$$

Proof: $\mathbf{x} \in I_i(\lambda) \Longrightarrow \phi_i(\mathbf{x}, \mathbf{c}) = \phi_i^*$. Thus,

$$\mathrm{Var}\,[\phi_i(\mathbf{X}, \mathbf{c}) \mid \mathbf{X} \in I_i(\lambda)] = \mathcal{E}\,[\{\phi_i(\mathbf{X}, \mathbf{c})\}^2 \mid \mathbf{X} \in I_i(\lambda)] - \{\mathcal{E}\,[\phi_i(\mathbf{X}, \mathbf{c}) \mid \mathbf{X} \in I_i(\lambda)]\}^2 = (\phi_i^*)^2 - (\phi_i^*)^2 = 0 \ .$$

Likewise, $\mathbf{x} \in I_{\neg i \neg\chi_i}(\lambda) \Longrightarrow \mathbf{x} \in I_{\neg\chi_i}(\lambda) \Longrightarrow \phi_i(\mathbf{x}, \mathbf{c}) = \phi_i(\mathbf{c}, \mathbf{c})$, which is also independent of $\mathbf{x}$. ■

The second lemma is a general result of mathematical statistics. although it is not found in many standard references (e.g. [2, 41. 53]). It is used frequently in the sequel for obtaining conditional variances.

Lemma 5.3.2 (Decomposition of (non-central) conditional second moment) *Let X be a random variable with space $\mathcal{S}$, $f : \mathcal{S} \longrightarrow \mathbb{R}$. and $k \in \mathbb{R}$. Also, let $A \subseteq \mathcal{S}$ such that $\mu_{f|A} \triangleq \mathcal{E}\left[f(X) \mid X \in A\right]$ and $Var\left[f(X) \mid X \in A\right] \triangleq \mathcal{E}\left[(f(X) - \mu_{f|A})^2 \mid X \in A\right]$ exist. Then*

$$\mathcal{E}\left[(f(X) - k)^2 \mid X \in A\right] = Var\left[f(X) \mid X \in A\right] + (\mu_{f|A} - k)^2 \quad .$$

Proof: By the linearity of the expected value operator. $\mathcal{E}\left[f(X) - \mu_{f|A} \mid X \in A\right] = \mu_{f|A} - \mu_{f|A} = 0$, so

$$\begin{aligned}
&\mathrm{Var}\left[f(X) \mid X \in A\right] + (\mu_{f|A} - k)^2 \\
&= \mathcal{E}\left[(f(X) - \mu_{f|A})^2 \mid X \in A\right] + (\mu_{f|A} - k)^2 \\
&= \mathcal{E}\left[(f(X) - \mu_{f|A})^2 \mid X \in A\right] + 2(\mu_{f|A} - k) \cdot \mathcal{E}\left[f(X) - \mu_{f|A} \mid X \in A\right] + (\mu_{f|A} - k)^2 \\
&= \mathcal{E}\left[(f(X) - \mu_{f|A})^2 + 2(f(X) - \mu_{f|A})(\mu_{f|A} - k) + (\mu_{f|A} - k)^2 \mid X \in A\right] \\
&= \mathcal{E}\left[\left\{(f(X) - \mu_{f|A}) + (\mu_{f|A} - k)\right\}^2 \mid X \in A\right] \\
&= \mathcal{E}\left[(f(X) - k)^2 \mid X \in A\right] \quad .
\end{aligned}$$

■

5.3.2 Central Moments of Unconditional Fitness Distributions. In the following theorem, decompositions of the first two central moments of a particular subfunction's contribution to an individual's fitness are obtained. For a subfunction i. each quantity is expressed as a linear combination of three appropriate conditional expectations: that given that the individual contains building block i, that given that the individual does not disrupt the competitive template with respect to subfunction i. and that given that neither of these conditions holds. These decompositions are useful in the analysis of the fitness distributions associated with both types of individuals.

Theorem 5.3.3 (Subfunction contribution expectations) *Let* $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ *be a separable fitness function,* $\mathbf{c} \in I_F$*, and* $\mathbf{X} \sim U(\mathcal{S})$ *where* $\mathcal{S} \subseteq I(\lambda)$*. Then the expected value of* $\phi_i(\mathbf{X}, \mathbf{c})$ *is*

$$\begin{aligned} \mu_i(\mathcal{S}, \mathbf{c}) \triangleq \; card\,(\mathcal{S})^{-1} \cdot \Big\{ & \qquad\quad \phi_i^* \cdot card\,(\mathcal{S} \cap I_i(\lambda)) \\ & + \mu_i^-(\mathcal{S}, \mathbf{c}) \cdot card\,(\mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})) \\ & + \phi_i(\mathbf{c}, \mathbf{c}) \cdot card\,(\mathcal{S} \cap I_{\neg i \neg \chi_i}(\lambda, \mathbf{c})) \Big\} , \end{aligned} \tag{13}$$

where

$$\mu_i^-(\mathcal{S}, \mathbf{c}) \triangleq \mathcal{E}\,[\phi_i(\mathbf{X}, \mathbf{c}) \mid \mathbf{X} \in \mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})] \; . \tag{14}$$

Also, the variance of $\phi_i(\mathbf{X}, \mathbf{c})$ *is*

$$\begin{aligned} \sigma_i^2(\mathcal{S}, \mathbf{c}) \triangleq \; \frac{1}{\mathcal{S}} \cdot \Big\{ & \qquad\qquad [\phi_i^* - \mu_i(\mathcal{S}, \mathbf{c})]^2 \cdot card\,(\mathcal{S} \cap I_i(\lambda)) \\ & + \left(\sigma_i^{2-}(\mathcal{S}, \mathbf{c}) + [\mu_i^-(\mathcal{S}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})]^2\right) \cdot card\,(\mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})) \\ & + \qquad [\phi_i(\mathbf{c}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})]^2 \cdot card\,(\mathcal{S} \cap I_{\neg i \neg \chi_i}(\lambda, \mathbf{c})) \Big\} . \end{aligned} \tag{15}$$

where

$$\sigma_i^{2-}(\mathcal{S}, \mathbf{c}) \triangleq Var\,[\phi_i(\mathbf{X}, \mathbf{c}) \mid \mathbf{X} \in \mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})] \; . \tag{16}$$

Proof: By definition, the expected fitness contribution of subfunction i is[4]

$$\mathcal{E}\left[\phi_i(\mathbf{X},\mathbf{c})\right] \triangleq \sum_{\mathbf{x}\in I(\lambda)} \phi_i(\mathbf{x},\mathbf{c}) \cdot \Pr[\mathbf{X}=\mathbf{x}] \ .$$

For $\mathbf{X} \sim U(\mathcal{S})$, this may be written as

$$\begin{aligned}
\mathcal{E}\left[\phi_i(\mathbf{X},\mathbf{c})\right] &= \operatorname{card}(\mathcal{S})^{-1} \cdot \Bigg\{ \sum_{\mathbf{x}\in\mathcal{S}\cap I_i(\lambda)} \phi_i(\mathbf{x},\mathbf{c}) \\
&\qquad + \sum_{\mathbf{x}\in\mathcal{S}\cap I_{\neg i\chi_i}(\lambda,\mathbf{c})} \phi_i(\mathbf{x},\mathbf{c}) \\
&\qquad + \sum_{\mathbf{x}\in\mathcal{S}\cap I_{\neg i\neg\chi_i}(\lambda,\mathbf{c})} \phi_i(\mathbf{x},\mathbf{c}) \Bigg\} \\
&= \operatorname{card}(\mathcal{S})^{-1} \Bigg\{ \phi_i^* \cdot \operatorname{card}(\mathcal{S}\cap I_i(\lambda)) \\
&\qquad + \mathcal{E}\left[\phi_i(\mathbf{X},\mathbf{c}) \mid \mathbf{X}\in\mathcal{S}\cap I_{\neg i\chi_i}(\lambda,\mathbf{c})\right] \cdot \operatorname{card}(\mathcal{S}\cap I_{\neg i\chi_i}(\lambda,\mathbf{c})) \\
&\qquad + \phi_i(\mathbf{c},\mathbf{c}) \cdot \operatorname{card}(\mathcal{S}\cap I_{\neg i\neg\chi_i}(\lambda,\mathbf{c})) \Bigg\} \\
&= \mu_i(\mathcal{S},\mathbf{c}) \ ,
\end{aligned}$$

where we have used the facts that

$$\mathbf{x}\in\mathcal{S}\cap I_i(\lambda) \Longrightarrow \phi_i(\mathbf{x},\mathbf{c}) = \phi_i^*$$

and

$$\mathbf{x}\in\mathcal{S}\cap I_{\neg i\neg\chi_i}(\lambda,\mathbf{c}) \Longrightarrow \phi_i(\mathbf{x},\mathbf{c}) = \phi_i(\mathbf{c},\mathbf{c}). \ .$$

[4] The expectation is taken over *all* individuals, whether or not they are fully specified with respect to subfunction i. This is not necessarily the same as either the expectation over all individuals which are fully specified with respect to subfunction i or the expectation over all length λ individuals which are fully specified with respect to subfunction i. The latter are equivalent, and also equivalent to

$$\mathcal{E}\left[\phi_i(\mathbf{A},\mathbf{L})\right] \triangleq \sum_{(\mathbf{a},\mathbf{l})\in\mathcal{A}^{k_i}\times\pi(\mathcal{L}_i)} \phi_i(\mathbf{a},\mathbf{l}) \cdot \Pr[\mathbf{A}=\mathbf{a}\wedge\mathbf{L}=\mathbf{l}] \ ,$$

which might be referred to as the *subfunction mean.*

Similarly, the variance of $\phi_i(\mathbf{X}, \mathbf{c})$ is by definition[5]

$$\mathrm{Var}\,[\phi_i(\mathbf{X}, \mathbf{c})] \;\triangleq\; \sum_{\mathbf{x} \in \mathcal{S}} \Big\{ \phi_i(\mathbf{x}, \mathbf{c}) - \mathcal{E}\,[\phi_i(\mathbf{X}, \mathbf{c})] \Big\}^2 \cdot \Pr[\mathbf{X} = \mathbf{x}] \;,$$

and for $\mathbf{X} \sim U(\mathcal{S})$,

$$\begin{aligned}
&\mathrm{Var}\,[\phi_i(\mathbf{X}, \mathbf{c})] \\
&= \sum_{\mathbf{x} \in \mathcal{S}} [\phi_i(\mathbf{x}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})]^2 \cdot \Pr[\mathbf{X} = \mathbf{x}] \\
&= \frac{1}{\mathcal{S}} \cdot \Big\{ \sum_{\mathbf{x} \in \mathcal{S} \cap I_i(\lambda)} [\phi_i(\mathbf{x}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})]^2 \\
&\qquad + \sum_{\mathbf{x} \in \mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})} [\phi_i(\mathbf{x}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})]^2 \\
&\qquad + \sum_{\mathbf{x} \in \mathcal{S} \cap I_{\neg i \neg \chi_i}(\lambda, \mathbf{c})} [\phi_i(\mathbf{x}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})]^2 \Big\} \\
&= \frac{1}{\mathcal{S}} \cdot \Big\{ [\phi_i^* - \mu_i(\mathcal{S}, \mathbf{c})]^2 \cdot \mathrm{card}\,(\mathcal{S} \cap I_i(\lambda)) \\
&\qquad + \mathcal{E}\,[\{\phi_i(\mathbf{X}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})\}^2 \mid \mathbf{X} \in \mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})] \cdot \mathrm{card}\,(\mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})) \\
&\qquad + [\phi_i(\mathbf{c}, \mathbf{c}) - \mu_i(\mathcal{S}, \mathbf{c})]^2 \cdot \mathrm{card}\,(\mathcal{S} \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})) \Big\} \;.
\end{aligned}$$

Upon simplification using Lemma 5.3.2, the result follows immediately. ■

The constants ϕ_i^* depend on the fitness function, as do the $\mu_i^-(\mathcal{S}, \mathbf{c})$'s, $\sigma_i^{2-}(\mathcal{S}, \mathbf{c})$'s, and the $\phi_i(\mathbf{c}, \mathbf{c})$'s, which each also depends on the competitive template $\mathbf{c}$. For the cases $\mathcal{S} = I(\lambda)$, $\mathcal{S} = I_\beta(\lambda)$, and $\mathcal{S} = I_{\neg\beta}(\lambda)$, analytical expressions for the cardinalities of $\mathcal{S} \cap I_i(\lambda)$, $\mathcal{S} \cap I_{\neg i \chi_i}(\lambda)$, and $\mathcal{S} \cap I_{\neg i \neg \chi_i}(\lambda)$ are given in Appendix A.

[5] Analogous to the preceding remark, the expectation is taken over the set of *all* individuals of length λ. Again, for $\lambda \neq \ell$, this is not necessarily the same as the expectation over all individuals which are fully specified with respect to subfunction i, whether or not restricted to those individuals of length λ. The latter expectations are both equivalent to

$$\mathrm{Var}\,[\phi_i(\mathbf{A}, \mathbf{L})] \triangleq \sum_{(\mathbf{a}, \mathbf{l}) \in \mathcal{A}^{k_i} \times \pi(\mathcal{L}_i)} \Big\{ \phi_i(\mathbf{a}, \mathbf{l}) - \mathcal{E}\,[\phi_i(\mathbf{A}, \mathbf{L})] \Big\}^2 \cdot \Pr[\mathbf{A} = \mathbf{a} \wedge \mathbf{L} = \mathbf{l}] \;,$$

which might be referred to as the *subfunction variance*.

The next theorem considers the specific fitness distributions associated with random individuals drawn uniformly from the classes I_β and $I_{\neg\beta}$. The following corollaries present decompositions of the first two central moments of each of these fitness distributions. For the case of normally distributed fitnesses, the first two moments determine the distribution. The first corollary relates to the fitness distribution for the class I_β.

Corollary 5.3.4 (Fitness expectations of individuals containing building block) *Let $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ be a separable fitness function, $\mathbf{c} \in I_F$, and $\mathbf{X} \sim U(I_\beta(\lambda))$. Then the expected value of $\Phi_c(\mathbf{X})$ is*

$$\mathcal{E}\ [\Phi_\mathbf{c}(\mathbf{X})] \quad = \quad \phi^*_\beta + \sum_{i \neq \beta} \mu_{i|\beta}(\lambda.\mathbf{c}) \quad ,$$

where

$$\mu_{i|\beta}(\lambda, \mathbf{c}) \quad \triangleq \quad \frac{1}{N_\beta(\lambda)} \left\{ \phi^*_i \cdot N_{\beta.i}(\lambda) + \mu^-_i(I_\beta(\lambda), \mathbf{c}) \cdot N_{\beta.\neg i\chi_i}(\lambda, \mathbf{c}) + \phi_i(\mathbf{c}.\mathbf{c}) \cdot N_{\beta.\neg i \neg\chi_i}(\lambda, \mathbf{c}) \right\} \quad . \tag{17}$$

and the $\mu^-_i(I_\beta(\lambda).\mathbf{c})$'s are defined by Equation 14. Furthermore, the variance of $\Phi_\mathbf{c}(\mathbf{X})$ is

$$\begin{aligned} Var\ [\Phi_\mathbf{c}(\mathbf{X})] \quad = \quad & \frac{1}{N_\beta(\lambda)} \sum_{i \neq \beta} \Bigg\{ \qquad [\phi^*_i - \mu_{i|\beta}(\lambda.\mathbf{c})]^2 \quad \cdot \quad N_{\beta,i}(\lambda) \\ & \quad + \quad \left(\sigma^{2-}_i(I_\beta(\lambda).\mathbf{c}) + [\mu^-_i(I_\beta(\lambda), \mathbf{c}) - \mu_{i|\beta}(\lambda.\mathbf{c})]^2 \right) \quad \cdot \quad N_{\beta,\neg i\chi_i}(\lambda.\mathbf{c}) \\ & \quad + \quad [\phi_i(\mathbf{c}, \mathbf{c}) - \mu_{i|\beta}(\lambda.\mathbf{c})]^2 \quad \cdot \quad N_{\beta,\neg i \neg\chi_i}(\lambda, \mathbf{c}) \qquad \Bigg\} \\ & + 2 \sum\sum_{i<j, i\neq\beta. j\neq\beta} Cov\ [\phi_i(\mathbf{X}, \mathbf{c}). \phi_j(\mathbf{X}, \mathbf{c}))] \quad , \end{aligned} \tag{18}$$

where the $\sigma^{2-}_i(I_\beta(\lambda), \mathbf{c})$'s are defined by Equation 16.

Proof: By the linearity of the expected value operator,

$$\mathcal{E}\ [\Phi_\mathbf{c}(\mathbf{X})] \quad = \quad \sum_{i=1}^{m} \mathcal{E}\ [\phi_i(\mathbf{X}, \mathbf{c})] \quad . \tag{19}$$

Because $\mathbf{X} \in I_\beta(\lambda)$, the term corresponding to $i = \beta$ is just $\mathcal{E}\ [\phi_\beta(\mathbf{X}, \mathbf{c})] = \phi_\beta^*$, while by Theorem 5.3.3, the terms corresponding to $i \neq \beta$ may be written

$$\begin{aligned} \mathcal{E}\ [\phi_i(\mathbf{X}, \mathbf{c})] &= \frac{1}{\operatorname{card}(I_\beta(\lambda))} \Big\{ \quad \phi_i^* \cdot \operatorname{card}(I_\beta(\lambda) \cap I_i(\lambda)) \\ &\qquad + \mu_i^-(I_\beta(\lambda), \mathbf{c}) \cdot \operatorname{card}(I_\beta(\lambda) \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})) \\ &\qquad + \phi_i(\mathbf{c}, \mathbf{c}) \cdot \operatorname{card}(I_\beta(\lambda) \cap I_{\neg i \neg \chi_i}(\lambda, \mathbf{c})) \Big\} \\ &= \mu_{i|\beta}(\lambda, \mathbf{c}) \ , \end{aligned}$$

which completes the proof of the claimed expected value. Similarly, the conditional variance may be expressed

$$\begin{aligned} \operatorname{Var}\ [\Phi_c(\mathbf{X})] &= \sum_{i=1}^{m} \operatorname{Var}\ [\phi_i(\mathbf{X}, \mathbf{c})] + 2 \sum\sum_{i<j} \operatorname{Cov}\ [\phi_i(\mathbf{X}, \mathbf{c}), \phi_j(\mathbf{X}, \mathbf{c}))] \\ &= \sum_{i \neq \beta} \operatorname{Var}\ [\phi_i(\mathbf{X}, \mathbf{c})] + 2 \sum\sum_{i<j, i\neq\beta, j\neq\beta} \operatorname{Cov}\ [\phi_i(\mathbf{X}, \mathbf{c}), \phi_j(\mathbf{X}, \mathbf{c}))] \ , \end{aligned}$$

where we have used Lemma 5.3.1. Theorem 5.3.3 implies that the remaining variance terms are

$$\begin{aligned} \operatorname{Var}\ [\phi_i(\mathbf{X}, \mathbf{c})] = \frac{1}{\operatorname{card}(I_\beta(\lambda))} \cdot \\ \Big\{ \quad & [\phi_i^* - \mu_i(I_\beta(\lambda), \mathbf{c})]^2 \cdot \operatorname{card}(I_\beta(\lambda) \cap I_i(\lambda)) \\ + & \left(\sigma_i^{2-}(I_\beta(\lambda), \mathbf{c}) + [\mu_i^-(I_\beta(\lambda), \mathbf{c}) - \mu_i(I_\beta(\lambda), \mathbf{c})]^2\right) \cdot \operatorname{card}(I_\beta(\lambda) \cap I_{\neg i \chi_i}(\lambda, \mathbf{c})) \\ + & [\phi_i(\mathbf{c}, \mathbf{c}) - \mu_i(I_\beta(\lambda), \mathbf{c})]^2 \cdot \operatorname{card}(I_\beta(\lambda) \cap I_{\neg i \neg \chi_i}(\lambda)) \Big\} \ . \end{aligned}$$

It remains only to note that $\mu_i(I_\beta(\lambda), \mathbf{c}) = \mu_{i|\beta}(\lambda, \mathbf{c})$. ■

The next corollary relates to the fitness distribution for the class $I_{\neg\beta}$.

Corollary 5.3.5 (Fitness expectations of individuals lacking building block) *Let* $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ *be a separable fitness function,* $\mathbf{c} \in I_F$, *and* $\mathbf{X} \sim U(I_{\neg\beta}(\lambda))$. *Then the expected value of* $\Phi_c(\mathbf{X})$ *is*

$$\mathcal{E}\ [\Phi_c(\mathbf{X})] = \sum_{i=1}^{m} \mu_{i|\neg\beta}(\lambda, \mathbf{c}) \ ,$$

where

$$\mu_{i|\neg\beta}(\lambda.\mathbf{c}) \quad \triangleq \quad \frac{1}{N_{\neg\beta}(\lambda)}\left\{\phi_i^* \cdot N_{\neg\beta,i}(\lambda) + \mu_i^-(I_{\neg\beta}(\lambda),\mathbf{c}) \cdot N_{\neg\beta,\neg i\chi_i}(\lambda,\mathbf{c}) + \phi_i(\mathbf{c},\mathbf{c}) \cdot N_{\neg\beta,\neg i\neg\chi_i}(\lambda,\mathbf{c})\right\} \quad .$$

and the $\mu_i^-(I_{\neg\beta}(\lambda),\mathbf{c})$'s are defined by Equation 14. Furthermore, the variance of $\Phi_{\mathbf{c}}(\mathbf{X})$ is

$$\begin{aligned} & Var\ [\Phi_{\mathbf{c}}(\mathbf{X})] = \\ & \quad \frac{1}{N_{\neg\beta}(\lambda)} \cdot \sum_{i=1}^{m} \Bigg\{ \qquad [\phi_i^* - \mu_{i|\neg\beta}(\lambda,\mathbf{c})]^2 \cdot N_{\neg\beta,i}(\lambda) \\ & \qquad + \left(\sigma_i^{2-}(I_{\neg\beta}(\lambda),\mathbf{c}) + [\mu_i^-(I_{\neg\beta}(\lambda),\mathbf{c}) - \mu_{i|\neg\beta}(\lambda.\mathbf{c})]^2\right) \cdot N_{\neg\beta,\neg i\chi_i}(\lambda,\mathbf{c}) \\ & \qquad + \qquad [\phi_i(\mathbf{c},\mathbf{c}) - \mu_{i|\neg\beta}(\lambda,\mathbf{c})]^2 \cdot N_{\neg\beta,\neg i\neg\chi_i}(\lambda,\mathbf{c}) \quad \Bigg\} \\ & \quad + 2\sum_{i<j}\sum Cov\ [\phi_i(\mathbf{X},\mathbf{c}),\phi_j(\mathbf{X},\mathbf{c}))] \quad , \end{aligned} \tag{20}$$

where the $\sigma_i^{2-}(I_{\neg\beta}(\lambda),\mathbf{c})$'s are defined by Equation 16.

Proof: Equation 19 holds, and by Theorem 5.3.3, each term may be written

$$\begin{aligned} \mathcal{E}\ [\phi_i(\mathbf{X},\mathbf{c})] \quad &= \quad \frac{1}{\text{card}\,(I_{\neg\beta}(\lambda))}\Bigg\{ \qquad \phi_i^* \cdot \text{card}\,(I_{\neg\beta}(\lambda) \cap I_i(\lambda)) \\ & \qquad + \mu_i^-(I_{\neg\beta}(\lambda),\mathbf{c}) \cdot \text{card}\,(I_{\neg\beta}(\lambda) \cap I_{\neg i\chi_i}(\lambda,\mathbf{c})) \\ & \qquad + \qquad \phi_i(\mathbf{c},\mathbf{c}) \cdot \text{card}\,(I_{\neg\beta}(\lambda) \cap I_{\neg i\neg\chi_i}(\lambda.\mathbf{c})) \quad \Bigg\} \\ &= \quad \mu_{i|\neg\beta}(\lambda,\mathbf{c}) \quad , \end{aligned}$$

which completes the proof of the claimed expected value.[6] Similarly, the conditional variance may be expressed

$$\text{Var}\ [\Phi_c(\mathbf{X})] \;=\; \sum_{i=1}^{m} \text{Var}\ [\phi_i(\mathbf{X}, \mathbf{c})] + 2 \sum_{i<j} \sum \text{Cov}\ [\phi_i(\mathbf{X}, \mathbf{c}), \phi_j(\mathbf{X}, \mathbf{c}))] \ .$$

By Theorem 5.3.3 each variance term may be written

$$\begin{aligned} \text{Var}\ [\phi_i(\mathbf{X}, \mathbf{c})] = \frac{1}{\text{card}\,(I_{\neg\beta}(\lambda))} \cdot \\ \Big\{ \qquad\qquad & [\phi_i^* - \mu_i(I_{\neg\beta}(\lambda), \mathbf{c})]^2 \quad \cdot \quad \text{card}\,(I_{\neg\beta}(\lambda) \cap I_i(\lambda)) \\ + \ \Big(\sigma_i^{2-}(I_{\neg\beta}(\lambda), \mathbf{c}) + [\mu_i^-(I_{\neg\beta}(\lambda), \mathbf{c}) - & \mu_i(I_{\neg\beta}(\lambda), \mathbf{c})]^2\Big) \quad \cdot \quad \text{card}\,(I_{\neg\beta}(\lambda) \cap I_{\neg i\chi_i}(\lambda, \mathbf{c})) \\ + \qquad\qquad [\phi_i(\mathbf{c}, \mathbf{c}) - & \mu_i(I_{\neg\beta}(\lambda), \mathbf{c})]^2 \quad \cdot \quad \text{card}\,(I_{\neg\beta}(\lambda) \cap I_{\neg i\neg\chi_i}(\lambda)) \qquad \Big\} \ . \end{aligned}$$

Since $\mu_i(I_{\neg\beta}(\lambda), \mathbf{c}) = \mu_{i|\neg\beta}(\lambda, \mathbf{c})$, the proof is complete. ■

5.3.3 Central Moments of Conditional Fitness Distributions. This section considers the conditional fitness distribution of individuals $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$, given that $\mathbf{X_1}$ shares λ_c defining loci with an individual $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$. It also considers the corresponding conditional fitness distribution for $\mathbf{X_2}$. The following lemma is important in the analysis of both distributions.

Lemma 5.3.6 (Number of individuals sharing λ_c defining loci — Part I)

Let $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ be a separable fitness function, $\mathcal{L}_\beta$ the set of defining loci of subfunction β, and $k \triangleq$ card $(\mathcal{L}_\beta)$. Suppose that $\mathbf{x_1} \in I_\beta(\lambda_1)$, where I is an lfGA individual space with finite genic alphabet $\mathcal{A}$ and nominal string length ℓ. Then the number of individuals $\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)$ sharing λ_c defining loci with

[6] As reflected in the expressions given in Appendix A, the case for which $i = \beta$ has

$$I_{\neg\beta}(\lambda) \cap I_i(\lambda) = \{\} \Longrightarrow N_{\neg\beta, i}(\lambda) = 0 \ ,$$

$$I_{\neg\beta}(\lambda) \cap I_{\neg i\chi_i}(\lambda) = I_{\neg i\chi_i}(\lambda) = I_{\neg\beta\chi_\beta}(\lambda) \Longrightarrow N_{\neg\beta, \neg i\chi_i}(\lambda) = N_{\neg\beta\chi_\beta}(\lambda) \ ,$$

and

$$I_{\neg\beta}(\lambda) \cap I_{\neg i\neg\chi_i}(\lambda) = I_{\neg i\neg\chi_i}(\lambda) = I_{\neg\beta\neg\chi_\beta}(\lambda) \Longrightarrow N_{\neg\beta, \neg i\neg\chi_i}(\lambda) = N_{\neg\beta\neg\chi_\beta}(\lambda) \ .$$

$\mathbf{x_1}$ *is*

$$N_c^{(\beta)}(\lambda_1, \lambda_2, \lambda_c) = [card\,(\mathcal{A})]^{\lambda_2} \binom{\lambda_1}{\lambda_c} \binom{\ell - \lambda_1}{\lambda_2 - \lambda_c} - [card\,(\mathcal{A})]^{\lambda_2 - k} \binom{\lambda_1 - k}{\lambda_c - k} \binom{\ell - \lambda_1}{\lambda_2 - \lambda_c} \quad . \tag{21}$$

Also, the number of pairs $(\mathbf{x_1}, \mathbf{x_2}) \in I_\beta(\lambda_1) \times I_{\neg\beta}(\lambda_2)$ *sharing* λ_c *defining loci is*

$$card\,(\omega(\lambda_1, \lambda_2, \lambda_c)) = N_\beta(\lambda_1) \cdot N_c^{(\beta)}(\lambda_1, \lambda_2, \lambda_c) \quad . \tag{22}$$

Proof: The sets $I_\beta(\lambda_2)$ and $I_{\neg\beta}(\lambda_2)$ form a partition of $I(\lambda_2)$. Thus,

$$\begin{aligned} &\text{card}\,(\{\mathbf{x_2} \in I_{\neg\beta}(\lambda_2) : \Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c\}) \\ &\quad = \text{card}\,(\{\mathbf{x_2} \in I(\lambda_2) : \Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c\}) - \text{card}\,(\{\mathbf{x_2} \in I_\beta(\lambda_2) : \Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c\}) \quad . \end{aligned}$$

For arbitrary individuals $\mathbf{x_2} \in I(\lambda_2)$, each of the λ_2 alleles has card $(\mathcal{A})$ possible values. For such individuals having λ_c defining loci in common with an individual $\mathbf{x_1} \in I_\beta(\lambda_1)$, the λ_c common loci must be chosen from the λ_1 loci of $\mathbf{x_1}$, while the remaining $\lambda_2 - \lambda_c$ loci of $\mathbf{x_2}$ must be chosen from the $\ell - \lambda_1$ loci for which $\mathbf{x_1}$ does not contain genes. Thus,

$$\text{card}\,(\{\mathbf{x_2} \in I(\lambda_2) : \Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c\}) = [\text{card}\,(\mathcal{A})]^{\lambda_2} \binom{\lambda_1}{\lambda_c} \binom{\ell - \lambda_1}{\lambda_2 - \lambda_c} \quad .$$

For arbitrary individuals $\mathbf{x_2} \in I_\beta(\lambda_2)$, each of the k alleles corresponding to the loci of subfunction β is fixed. The remaining $\lambda_2 - k$ alleles have card $(\mathcal{A})$ possible values each. For such individuals having λ_c defining loci in common with an individual $\mathbf{x_1} \in I_\beta(\lambda_1)$, k of the λ_c common loci are those of subfunction β. The remaining $\lambda_c - k$ must be chosen from the other $\lambda_1 - k$ loci of $\mathbf{x_1}$. The remaining $\lambda_2 - \lambda_c$ loci of $\mathbf{x_2}$ must be chosen from the $\ell - \lambda_1$ loci for which $\mathbf{x_1}$ does not contain genes. Thus,

$$\mathrm{card}\left(\{\mathbf{x_2} \in I_\beta(\lambda_2) : \Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c\}\right) \;=\; [\mathrm{card}\,(\mathcal{A})]^{\lambda_2 - k} \binom{\lambda_1 - k}{\lambda_c - k} \binom{\ell - \lambda_1}{\lambda_2 - \lambda_c} \;.$$

which completes the proof of the claimed expression for $N_c^{(\beta)}(\lambda_1, \lambda_2, \lambda_c)$. Since $\mathbf{x_1}$ is arbitrary, every $\mathbf{x_1} \in I_\beta(\lambda_1)$ has the same number of individuals $\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)$ such that $(\mathbf{x_1}, \mathbf{x_2}) \in \omega(\lambda_1, \lambda_2, \lambda_c)$, which proves the claim regarding the cardinality of $\omega(\lambda_1, \lambda_2, \lambda_c)$. ■

The significance of Equation 21 is not so much the specific expression for $N_c^{(\beta)}(\lambda_1, \lambda_2, \lambda_c)$, but rather the fact that it is independent of $\mathbf{x_1}$. That is, for a given λ_c, every individual $\mathbf{x_1} \in I_\beta(\lambda_1)$ has the same number of individuals $\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)$ with which it shares λ_c defining loci. The following theorem is a consequence.

Theorem 5.3.7 (Conditional distribution of individuals containing building block)

Let $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ be a separable fitness function, and $\mathbf{c} \in I_F$. Suppose $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$ and $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$ are independent. Then the conditional distribution of $\mathbf{X_1}$ given that $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ is also $U(I_\beta(\lambda_1))$. That is, the probability that $\mathbf{X_1} = \mathbf{x_1}$ given that $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ is

$$f_1(\mathbf{x_1} \mid \Lambda_c = \lambda_c) \;\triangleq\; [N_\beta(\lambda_1)]^{-1} \;.$$

Proof: By the Law of Total Probability, the conditional probability that $\mathbf{X_1} = \mathbf{x_1}$ given that $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ is

$$\begin{aligned} f_1(\mathbf{x_1} \mid \Lambda_c = \lambda_c) &= \Pr[\mathbf{X_1} = \mathbf{x_1} \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c] \\ &= \sum_{\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)} \Pr[\mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2} \mid \Lambda_c = \lambda_c] \;. \end{aligned}$$

For $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$ and $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$ independent, this may be written

$$\begin{aligned}
&f_1(\mathbf{x_1} \mid \Lambda_c = \lambda_c) \\
&= \sum_{\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)} \frac{\Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c \mid \mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2}] \cdot \Pr[\mathbf{X_1} = \mathbf{x_1}] \cdot \Pr[\mathbf{X_2} = \mathbf{x_2}]}{\Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c]} \\
&= \frac{[N_\beta(\lambda_1)]^{-1} \cdot [N_{\neg\beta}(\lambda_2)]^{-1} \cdot \sum_{\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)} \Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c \mid \mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2}]}{\operatorname{card}(\omega(\lambda_1, \lambda_2, \lambda_c)) \cdot [N_\beta(\lambda_1)]^{-1} \cdot [N_{\neg\beta}(\lambda_2)]^{-1}} \quad .
\end{aligned}$$

Because $\Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c \mid \mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2}] = 1$ if $\Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c$, and 0 otherwise, this is just

$$\begin{aligned}
f_1(\mathbf{x_1} \mid \Lambda_c = \lambda_c) &= \operatorname{card}(\omega(\lambda_1, \lambda_2, \lambda_c))^{-1} \cdot N_c^{(\beta)}(\lambda_1, \lambda_2, \lambda_c) \\
&= [N_\beta(\lambda_1)]^{-1} \quad .
\end{aligned}$$

where we have used Lemma 5.3.6. ■

Of course, since the conditional distribution of $\mathbf{X_1}$ is identical to its unconditional distribution, the conditional expectations are identical to the unconditional expectations. In particular, the following corollary gives the conditional expectation and variance of the fitness distribution for the class $I_\beta(\lambda)$.

Corollary 5.3.8 (Conditional fitness expectations of individuals containing building block)

Let $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ *be a separable fitness function, and* $\mathbf{c} \in I_F$. *Suppose* $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$ *and* $\mathbf{X_2} \sim U(I_\beta(\lambda_2))$. *Then the conditional expectation of* $\Phi_c(\mathbf{X_1})$ *given that* $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ *is*

$$\mathcal{E}\,[\Phi_c(\mathbf{X_1}) \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c] \quad = \quad \phi_\beta^* + \sum_{i \neq \beta} \mu_{i|\beta}(\lambda_1, \mathbf{c}) \quad ,$$

where

$$\mu_{i|\beta}(\lambda_1, \mathbf{c}) \quad \triangleq \quad \frac{1}{N_\beta(\lambda_1)} \left\{ \phi_i^* \cdot N_{\beta.i}(\lambda_1) + \mu_i^-(I_\beta(\lambda_1), \mathbf{c}) \cdot N_{\beta.\neg i\chi_i}(\lambda_1, \mathbf{c}) + \phi_i(\mathbf{c}, \mathbf{c}) \cdot N_{\beta.\neg i \neg\chi_i}(\lambda_1, \mathbf{c}) \right\} \quad ,$$

and the $\mu_i^-(I_\beta(\lambda_1),\mathbf{c})$'s are defined by Equation 14. Furthermore, the conditional variance of $\Phi_\mathbf{c}(\mathbf{X_1})$ given that $\Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c$ is

$$\begin{aligned}
& Var\,[\Phi_\mathbf{c}(\mathbf{X_1}) \mid \Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c] \\
= \; & \frac{1}{N_\beta(\lambda_1)} \sum_{i \neq \beta} \Big\{ \quad [\phi_i^* - \mu_{i|\beta}(\lambda_1,\mathbf{c})]^2 \cdot N_{\beta,i}(\lambda_1) \\
& \quad + \left(\sigma_i^{2-}(I_\beta(\lambda_1).\mathbf{c}) + [\mu_i^-(\lambda_1.\mathbf{c}) - \mu_{i|\beta}(\lambda_1,\mathbf{c})]^2\right) \cdot N_{\beta,\neg i \chi_i}(\lambda_1.\mathbf{c}) \\
& \quad + [\phi_i(\mathbf{c},\mathbf{c}) - \mu_{i|\beta}(\lambda_1,\mathbf{c})]^2 \cdot N_{\beta,\neg i \neg \chi_i}(\lambda_1.\mathbf{c}) \quad \Big\} \\
& + 2 \sum\sum_{i<j, i\neq\beta, j\neq\beta} Cov\,[\phi_i(\mathbf{X_1}.\mathbf{c}).\phi_j(\mathbf{X_1}.\mathbf{c})] \quad .
\end{aligned} \tag{23}$$

where the $\sigma_i^{2-}(I_\beta(\lambda_1),\mathbf{c})$'s are defined by Equation 16.

Proof: The result follows immediately from Theorem 5.3.7 and Corollary 5.3.4. ■

The remainder of this section considers the conditional fitness distribution of individuals $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$, given that $\mathbf{X_2}$ shares λ_c defining loci with an individual $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$. In contrast to the situation for the class I_β, here the conditional distribution of fitnesses is not in general identical to the unconditional distribution. This is because for an individual $\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)$ and a given number λ_c of common defining loci, the number of individuals $\mathbf{x_1} \in I_\beta(\lambda_1)$ such that $\Lambda_c(\mathbf{x_1},\mathbf{x_2}) = \lambda_c$ depends on the choice of defining loci for $\mathbf{x_2}$. This is made precise by the following lemma.

Lemma 5.3.9 (Number of individuals sharing λ_c defining loci — Part II) *Let $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ be a separable fitness function, $\mathcal{L}_\beta$ the set of defining loci of subfunction β, and $k \triangleq card(\mathcal{L}_\beta)$. Suppose that $\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)$, where I is an lfGA individual space with finite genic alphabet $\mathcal{A}$ and nominal string length ℓ. Then the number of individuals $\mathbf{x_1} \in I_\beta(\lambda_1)$ sharing λ_c defining loci with $\mathbf{x_2}$ is*

$$N_c^{(\neg\beta)}(\lambda_1,\lambda_2,\lambda_c) \quad = \quad [card\,(\mathcal{A})]^{\lambda_1 - k} \binom{\lambda_2 - r(\mathbf{x_2})}{\lambda_c - r(\mathbf{x_2})} \binom{\ell - \lambda_2 - (k - r(\mathbf{x_2}))}{\lambda_1 - \lambda_c - (k - r(\mathbf{x_2}))} \quad . \tag{24}$$

where $r(\mathbf{x_2})$ is the number of loci of subfunction β with respect to which $\mathbf{x_2}$ is defined.

Proof: For arbitrary individuals $\mathbf{x_1} \in I_\beta(\lambda_1)$, each of the k alleles corresponding to the loci of subfunction β is fixed. The remaining $\lambda_1 - k$ alleles have card$(\mathcal{A})$ possible values each. For such individuals having λ_c defining loci in common with an individual $\mathbf{x_2} \in I_{\neg\beta}(\lambda_2)$, $r(\mathbf{x_2})$ of the λ_c common loci are those of subfunction β. The remaining $\lambda_c - r(\mathbf{x_2})$ must be chosen from the other $\lambda_2 - r(\mathbf{x_2})$ loci of $\mathbf{x_2}$. Of the $\lambda_1 - \lambda_c$ defining loci of $\mathbf{x_1}$ which are not shared by $\mathbf{x_2}$, $k - r(\mathbf{x_2})$ are those of subfunction β. The remaining $\lambda_1 - \lambda_c - (k - r(\mathbf{x_2}))$ must be chosen from the $\ell - \lambda_2 - (k - r(\mathbf{x_2})$ non–subfunction β loci for which $\mathbf{x_2}$ does not contain genes, which completes the proof. ■

The conditional moments of the fitness distribution for individuals $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$, given that $\mathbf{X_2}$ shares λ_c defining loci with an individual $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$. Because the number of individuals $\mathbf{x_1} \in I_\beta(\lambda_1)$ such that $\Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c$ (Equation 24) depends on $\mathbf{x_2}$, and in particular on its defining loci, the conditional distribution is not in general uniform. The following theorem presents the conditional probability density function for the class $I_{\neg\beta}$.

Theorem 5.3.10 (Conditional distribution of individuals lacking building block)

Let $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ *be a separable fitness function, and* $\mathbf{c} \in I_F$. *Suppose* $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$ *and* $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$ *are independent. Then the conditional density function of* $\mathbf{X_2}$ *given that* $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ *is*

$$f_2(\mathbf{x_2} \mid \Lambda_c = \lambda_c) \quad \triangleq f_{2,r} \quad .$$

where

$$\mathbf{x_2} \in R(r, \lambda_2) \quad \triangleq \quad \{\mathbf{x} \in I_{\neg\beta}(\lambda_2) : \mathbf{x} \textit{ is defined w.r.t. exactly } r \textit{ of the loci of subfunction } \beta\} \quad , \qquad (25)$$

$$f_{2,r} \quad \triangleq \quad N_c^{(\neg\beta)}(\lambda_1, \lambda_2, \lambda_c, r) \cdot card\,(\omega(\lambda_1, \lambda_2, \lambda_c))^{-1} \quad , \qquad (26)$$

the $N_c^{(\neg\beta)}(\lambda_1, \lambda_2, \lambda_c, r)$*'s are defined by Equation 24, and* $card\,(\omega(\lambda_1, \lambda_2, \lambda_c))$ *is given by Equation 22.*

Proof: By the Law of Total Probability, the conditional probability that $\mathbf{X_2} = \mathbf{x_2}$ given that $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ is

$$\begin{aligned} f_2(\mathbf{x_2} \mid \Lambda_c = \lambda_c) &= \Pr[\mathbf{X_2} = \mathbf{x_2} \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c] \\ &= \sum_{\mathbf{x_1} \in I_\beta(\lambda_1)} \Pr[\mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2} \mid \Lambda_c = \lambda_c] \ . \end{aligned}$$

For $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$ and $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$ independent, this may be written

$$\begin{aligned} & f_2(\mathbf{x_2} \mid \Lambda_c = \lambda_c) \\ &= \sum_{\mathbf{x_1} \in I_\beta(\lambda_1)} \frac{\Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c \mid \mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2}] \cdot \Pr[\mathbf{X_1} = \mathbf{x_1}] \cdot \Pr[\mathbf{X_2} = \mathbf{x_2}]}{\Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c]} \\ &= \frac{[N_\beta(\lambda_1)]^{-1} \cdot [N_{\neg\beta}(\lambda_2)]^{-1} \cdot \sum_{\mathbf{x_1} \in I_\beta(\lambda_1)} \Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c \mid \mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2}]}{\mathrm{card}\,(\omega(\lambda_1, \lambda_2, \lambda_c)) \cdot [N_\beta(\lambda_1)]^{-1} \cdot [N_{\neg\beta}(\lambda_2)]^{-1}} \ . \end{aligned}$$

Because $\Pr[\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c \mid \mathbf{X_1} = \mathbf{x_1} \wedge \mathbf{X_2} = \mathbf{x_2}] = 1$ if $\Lambda_c(\mathbf{x_1}, \mathbf{x_2}) = \lambda_c$, and 0 otherwise, this is just

$$f_2(\mathbf{x_2} \mid \Lambda_c = \lambda_c) = \mathrm{card}\,(\omega(\lambda_1, \lambda_2, \lambda_c))^{-1} \cdot N_c^{(\neg\beta)}(\lambda_1, \lambda_2, \lambda_c, r(\mathbf{x_2})) \ .$$

where we have used Lemma 5.3.9. ■

Because the conditional distribution of $\mathbf{X_2}$ is not uniform, the decomposition of the subfunction contributions provided by Theorem 5.3.3 does not apply directly. The following theorem presents a decomposition which does apply.

Theorem 5.3.11 (Conditional expectations of subfunction contributions) *Let* $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ *be a separable fitness function, and* $\mathbf{c} \in I_F$*. Suppose* $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$ *and* $\mathbf{X_2} \sim U(I_{\neg\beta}(\lambda_2))$

are independent. Then the conditional expectation of $\phi_i(\mathbf{X_2},\mathbf{c})$ *given that* $\Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c$ *is*

$$\mu_{i|\neg\beta\lambda_c}(\lambda_2,\mathbf{c}) \quad \triangleq \quad \sum_{r=0}^{k} card\,(R(r,\lambda_2)) \cdot f_{2,r} \cdot \mu_i(R(r,\lambda_2),\mathbf{c}) \quad , \tag{27}$$

where the $R(r,\lambda_2)$*'s are defined by Equation 25, the* $f_{2,r}$*'s are defined by Equation 26, and the* $\mu_i(R(r,\lambda_2),\mathbf{c})$*'s are defined by Equation 13. Also, the conditional variance of* $\phi_i(\mathbf{X_2},\mathbf{c})$ *is*

$$\sigma^2_{i|\neg\beta\lambda_c}(\lambda_2,\mathbf{c}) \quad = \quad \sum_{r=0}^{k} card\,(R(r,\lambda_2)) \cdot f_{2,r} \cdot \left[\sigma_i^2(R(r,\lambda_2),\mathbf{c}) + \{\mu_i(R(r,\lambda_2),\mathbf{c}) - \mu_{i|\neg\beta\lambda_c}(\lambda_2,\mathbf{c})\}^2\right] \quad . \tag{28}$$

where the $\sigma_i^2(R(r,\lambda_2),\mathbf{c})$*'s are defined by Equation 15.*

Proof: By definition, the conditional expectation of the fitness contribution of subfunction i is[7]

$$\mathcal{E}\,[\phi_i(\mathbf{X_2},\mathbf{c}) \mid \Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c] \quad \triangleq \quad \sum_{\mathbf{x_2}\in I_{\neg\beta}(\lambda_2)} \phi_i(\mathbf{x_2},\mathbf{c}) \cdot \Pr[\mathbf{X_2} = \mathbf{x_2} \mid \Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c] \quad .$$

Because $\{R(0,\lambda_2),\ldots,R(k,\lambda_2)\}$ is a partition of $I_{\neg\beta}(\lambda_2)$, this may be written as

$$\mathcal{E}\,[\phi_i(\mathbf{X_2},\mathbf{c}) \mid \Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c] \quad = \quad \sum_{r=0}^{k} \sum_{\mathbf{x_2}\in R(r,\lambda_2)} \phi_i(\mathbf{x_2},\mathbf{c}) \cdot \Pr[\mathbf{X_2} = \mathbf{x_2} \mid \Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c] \quad .$$

By Theorem 5.3.10, the conditional probability density of $\mathbf{X_2}$ is constant over each $R(r)$, so that

$$\begin{aligned}
\mathcal{E}\,[\phi_i(\mathbf{X_2},\mathbf{c}) \mid \Lambda_c(\mathbf{X_1},\mathbf{X_2}) = \lambda_c] \quad &= \quad \sum_{r=0}^{k} \sum_{\mathbf{x_2}\in R(r,\lambda_2)} \phi_i(\mathbf{x_2},\mathbf{c}) \cdot f_{2,r} \\
&= \quad \sum_{r=0}^{k} \left[f_{2,r} \sum_{\mathbf{x_2}\in R(r,\lambda_2)} \phi_i(\mathbf{x_2},\mathbf{c}) \right] \\
&= \quad \sum_{r=0}^{k} f_{2,r} \cdot \mathrm{card}\,(R(r,\lambda_2)) \cdot \mu_i(R(r,\lambda_2),\mathbf{c}) \quad ,
\end{aligned}$$

[7] See footnote 4.

which completes the proof of the claimed conditional expectation. Similarly, the conditional variance of $\phi_i(\mathbf{X_2}, \mathbf{c})$ is by definition[8]

$$\begin{aligned}
&\mathrm{Var}\left[\phi_i(\mathbf{X_2}, \mathbf{c}) \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c\right] \\
&\triangleq \sum_{\mathbf{x}\in \mathbf{I}_{\neg\beta}(\lambda_2)} \left\{ \phi_i(\mathbf{x_2}, \mathbf{c}) - \mathcal{E}\left[\phi_i(\mathbf{X_2}, \mathbf{c})\right] \right\}^2 \cdot \Pr[\mathbf{X_2} = \mathbf{x_2} \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c] \\
&= \sum_{r=0}^{k} \left[f_{2,r} \mathrm{card}\left(R(r, \lambda_2)\right) \cdot \mathcal{E}\left[\{\phi_i(\mathbf{X_2}, \mathbf{c}) - \mu_{i|\neg\beta\lambda_c}(\lambda_2, \mathbf{c})\}^2 \mid \mathbf{X_2} \in R(r, \lambda_2)\right]\right] \quad .
\end{aligned}$$

where we have again used the facts that the $R(r, \lambda_2)$'s form a partition of $I_{\neg\beta}(\lambda_2)$ and that the conditional density of $\mathbf{X_2}$ is constant over each $R(r, \lambda_2)$. Upon simplification using Lemma 5.3.2, the result follows immediately. ■

The following corollary presents the central moments of the conditional fitness distribution of the class $I_{\neg\beta}$.

Corollary 5.3.12 (Conditional fitness expectations of individuals lacking building block)

Let $\Phi \triangleq \sum_i \phi_i : I \times I_F \longrightarrow \mathbb{R}$ be a separable fitness function, and $\mathbf{c} \in I_F$. Suppose $\mathbf{X_1} \sim U(I_\beta(\lambda_1))$ and $\mathbf{X_2} \sim U(I_\beta(\lambda_2))$. Then the conditional expectation of $\Phi_c(\mathbf{X_2})$ given that $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ is

$$\mathcal{E}\left[\Phi_\mathbf{c}(\mathbf{X_2}) \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c\right] \quad = \quad \sum_{i=1}^{m} \mu_{i|\neg\beta,\lambda_c}(\lambda_2, \mathbf{c}) \quad ,$$

where the $\mu_{i|\neg\beta,\lambda_c}(\lambda_2, \mathbf{c})$'s are defined by Equation 27. Furthermore, the conditional variance of $\Phi_\mathbf{c}(\mathbf{X_2})$ given that $\Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c$ is

$$\begin{aligned}
&Var\left[\Phi_\mathbf{c}(\mathbf{X_2}) \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c\right] \\
&= \sum_{i=1}^{m} \sigma^2_{i|\neg\beta\lambda_c}(\lambda_2, \mathbf{c}) + 2 \sum\sum_{i<j} Cov\left[\phi_i(\mathbf{X_2}, \mathbf{c}), \phi_j(\mathbf{X_2}, \mathbf{c}) \mid \Lambda_c(\mathbf{X_1}, \mathbf{X_2}) = \lambda_c\right] \quad .
\end{aligned}$$

where the $\sigma^2_{i|\neg\beta\lambda_c}(\lambda_2, \mathbf{c})$'s are defined by Equation 28.

[8] See footnote 5

Proof: The results follow immediately from the linearity of the expected value operator and Theorem 4, §4.9, of Hogg and Craig [41], respectively. ■

5.4 Application of Tournament Selection Model

Under certain conditions, the model developed in Sections 5.1 and 5.2 enables exact determination of the expected state following binary tournament selection. Specifically, for certain classes of binary tournament selection algorithms (identified in Section 5.2), the model exactly predicts the probability that an individual in population $P(t)$ belongs to one of two classes. This section demonstrates the validity of the model. The application chosen for the demonstration is the prediction of the fraction of individuals which contain a particular building block in each generation of one selection episode of a fast messy genetic algorithm. The experimental design is discussed in Section 5.4.1, and the results are presented in Section 5.4.2.

5.4.1 Experimental Design. The fraction of individuals containing a particular building block in each generation is compared to the fraction predicted by the proposed model. The predicted mean and variance of the fitnesses of individuals containing the building block are also compared to the observed values, and similarly for individuals lacking the building block. The fast messy genetic algorithm is executed[9] ten times using different random seeds to facilitate statistically significant conclusions. Standard fast messy genetic algorithm parameters are used, including no thresholding in the first episode. Other relevant[10] parameters are presented in the remainder of this section, which discusses the fitness function and modeling assumptions used.

The fitness function for these experiments is that used by Goldberg et al. to demonstrate the feasibility of the fast messy genetic algorithm [35]. This function, a "tightly-coded 50-bit order-5 fully deceptive trap

[9] The experiments are performed on one node of an Intel Paragon using AFIT's fast messy genetic algorithm implementation [28], modified to collect statistics.

[10] Many of the fast messy genetic algorithm parameters have no impact on the experiments performed here. In particular, because no thresholding is used, the shuffle size is of no consequence. Also irrelevant are the cut and splice probabilities, the durations of the primordial and juxtapositional phases, and the filtering and thresholding parameters for other than the first episode.

function," as well as the underlying lfGA representation. is defined in the following in the notation of Section 2.6.1 and Definition 4.1.2.

The individual space is defined over genic alphabet $\mathcal{A} = \{0.1\}$, with nominal string length $\ell = 50$ and overflow factor $o = 1.6$, so that

$$I \quad = \quad \bigcup_{\lambda=0}^{80} (\{0,1\}^{\lambda} \times \{1.\ldots,50\}^{\lambda}) \quad ,$$

and the set of fully specified individuals is

$$I_F \triangleq I(50) = \{((a_1, \ldots, a_{50}), (l_1, \ldots . l_{50})) \in I : l_i = l_j \Longleftrightarrow i = j\} \quad .$$

The overlay mapping is $\Gamma : I \times I_F \longrightarrow \mathcal{A}^{\ell}$ as defined by Equation 3. For $i \in \{1, \ldots, 10\}$, take $\mathcal{L}_i = \{5i - 4, \ldots, 5i\}$. corresponding to a "tight" coding. so that the projection mappings $\mathcal{P}_{\mathcal{L}_i} : \{0.1\}^{50} \longrightarrow \{0,1\}^5$ are

$$\mathcal{P}_{\mathcal{L}_i}(a_1, \ldots, a_{50}) \quad \triangleq \quad (a_{5i-4}, \ldots, a_{5i}) \quad .$$

For $i \in \{1.\ldots.10\}$. the (identical) decoding subfunctions are $D_i = \hat{D}$ where $\hat{D} : \{0,1\}^5 \longrightarrow \mathbb{R}$ is the "counting ones" function

$$\hat{D}(a_1.\ldots, a_5) \quad \triangleq \quad \mathrm{card}\,(\{i \in \{1.\ldots, 5\} : a_i = 1\}) \quad .$$

Also for $i \in \{1.\ldots.10\}$, the (identical) objective subfunctions are $f_i = \hat{f}$ where $\hat{f} : \mathbb{R} \longrightarrow \mathbb{R}$ is the "trap" function

$$\hat{f}(x) \quad \triangleq \quad \begin{cases} 0.58(4 - x) & , \text{if } x \leq 4 \\ 1.00 & , \text{if } x > 4 \end{cases} \quad .$$

Finally, the fitness function $\Phi : I \times I_F \longrightarrow \mathbb{R}$ is

$$\Phi \triangleq \sum_{i=1}^{10} \phi_i \ ,$$

where each $\phi_i \triangleq f_i \circ D_i \circ \mathcal{P}_{\mathcal{L}_i} \circ \Gamma$ is a fitness subfunction. The competitive template for these experiments is $\mathbf{c} = ((0, \ldots, 0), (1, \ldots, 50))$, as used by Goldberg et al. [35]. The initial population of the fast messy genetic algorithm is drawn from a uniform distribution over $I(\ell - k)$, thus $\lambda = \ell' \triangleq \ell - k = 45$.

It is clear that $\Phi_{\mathbf{c}}$ is an order-5 separable lfGA fitness function. Thus, the decompositions of the central moments presented in Section 5.3 are applicable. The assumption of zero covariances throughout is somewhat justified, because the condition $\lambda \gg k$ results in near-independence of the subfunction contributions. It is then straightforward to obtain

$$\begin{aligned}
\phi_i^* &= 1 \ , \\
\phi_i(\mathbf{c}, \mathbf{c}) &= 0.58 \ , \\
\mu_i^-(I_\beta(\lambda), \mathbf{c}) &= 0.2471 \ , \text{for } i \neq \beta \ , \\
\sigma_i^{2-}(I_\beta(\lambda), \mathbf{c}) &= 0.0187 \ , \text{for } i \neq \beta \ , \\
\mu_i^-(I_{\neg\beta}(\lambda), \mathbf{c}) &= 0.2441 \ , \text{for } i \in \{1, \ldots, 10\} \ .
\end{aligned}$$

and

$$\sigma_i^{2-}(I_{\neg\beta}(\lambda), \mathbf{c}) = 0.0188 \ , \text{for } i \in \{1, \ldots, 10\} \ .$$

The values of $\mu_i^-(I_\beta(\lambda), \mathbf{c})$ and $\mu_i^-(I_{\neg\beta}(\lambda), \mathbf{c})$ are unmistakably similar, as are those of $\sigma_i^{2-}(I_\beta(\lambda), \mathbf{c})$ and $\sigma_i^{2-}(I_{\neg\beta}(\lambda), \mathbf{c})$. This similarity is due to the previously mentioned near-independence of the subfunction contributions.

The mean and variance of the fitness distribution for individuals containing a particular building block are thus 3.493 and 0.290, respectively. Similarly, the unconditional mean and variance for individuals lacking the building block are 2.730 and 0.5662, which are essentially identical to the corresponding conditional values (for all λ_c such that the conditional distribution exists).

In order to gain analytical tractability, the decision making model proposed in Section 5.1 neglects the possibility of ties by assuming that the fitnesses of individuals are random variables of the continuous type. For the fitness function employed in these experiments, this assumption does not hold. Thus, for purposes of the decision making model, the computational experiments reported here approximate the fitness distributions by normal distributions with the means and variances just calculated.

The expressions given in Section 5.1 for the probabilities of correct decision making also assume that the fitnesses of the ancestors of the competing individuals are mutually independent. This assumption holds provided that the ancestors are distinct. In the absence of thresholding, each individual possesses a maximum of 2^t ancestors. Thus, if all of $\mathbf{x_1}$'s ancestors are distinct, and likewise those of $\mathbf{x_2}$, then the probability in a finite population of size N that a specific ancestor of $\mathbf{x_1}$ is also an ancestor of $\mathbf{x_2}$ is $2^t N^{-1}$. Neglecting the statistical dependence of a particular individual's ancestors, the expected number of non-distinct ancestors is therefore $(2^t N^{-1})2^t = 4^t N^{-1}$. That is, less than one common ancestor is expected provided that $t < \log_4 N$, which is one half of the "takeover time." These experiments use a population size of $N = 1786$. On the basis of the preceding argument, it is reasonable to expect the probabilities of correct decision making to be accurate through generation $5 < \log_4 1786 \approx 5.4$.

Finally, after the first iteration of selection, closed form solutions for the probabilities of correct decision making do not exist. Consequently, these values are obtained numerically (see Appendix B).

5.4.2 Experimental Results. The predicted and observed fraction of individuals containing the building block in each generation is shown in Figure 24. The predicted state is accurate through the third generation, after which it becomes overly "optimistic." That is, it predicts a greater fraction of individuals containing the building block than is observed. The source of this over-optimism may be explained by

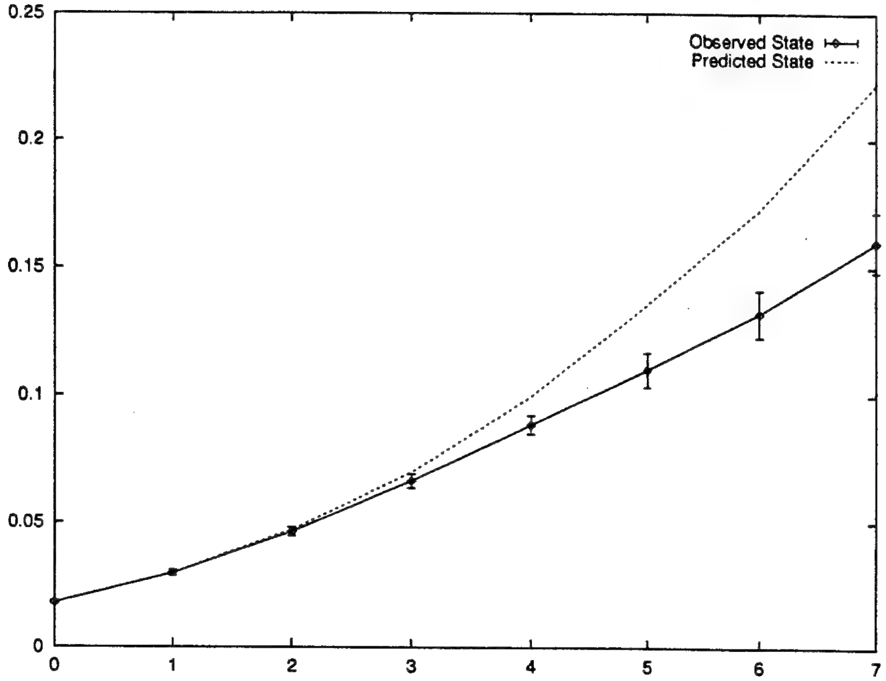


Figure 24. Predicted vs. Observed fmGA State

comparing the predicted and observed fitness distribution moments. The predicted and observed mean fitnesses for individuals containing the building block are shown in Figure 25. Again. the prediction is initially accurate. but after the fourth generation. it underestimates the actual mean. The inaccuracy in this prediction begins in a later generation than the inaccuracy in the state prediction. Thus. the former is an effect of. rather than a cause of. the latter inaccuracy. The mean fitness of individuals lacking the building block is shown in Figure 26. The results are qualitatively similar to those for individuals containing the building block. The predicted fitness standard deviations are compared in Figure 27 to their observed counterparts for individuals containing the building block. Although not as accurate as the prediction of the distribution means. the prediction for this statistic is well within the range of observed values. One notable difference between this result and those for the distribution means is the inaccuracy in the prediction for the initial population. This is attributable to the fact that the prediction neglects the covariances of the subfunction contributions, which although small, can easily be seen to be negative. The same remark applies to the predicted standard deviation of the fitnesses of those individuals lacking the building block, which is

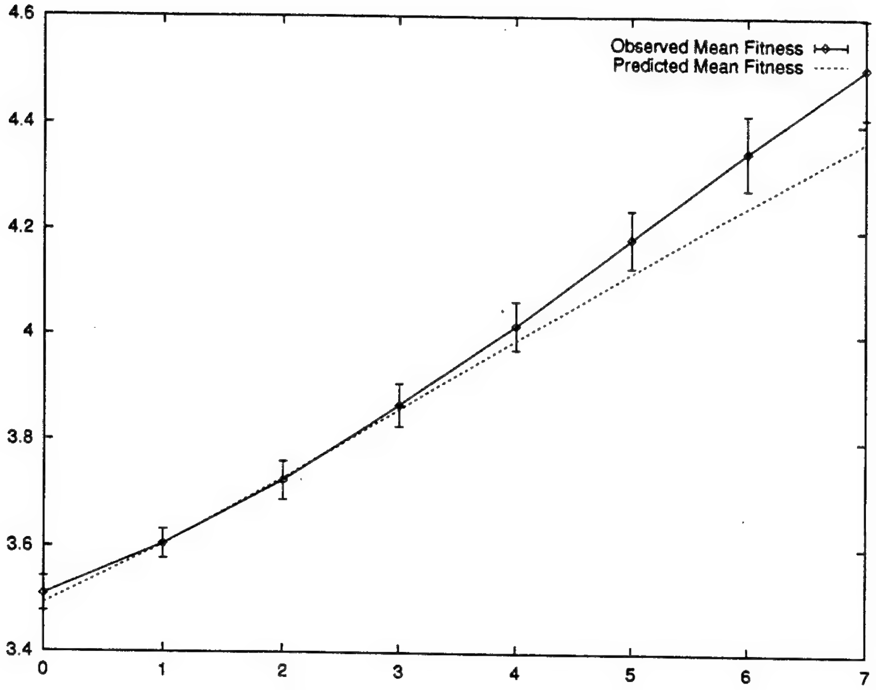


Figure 25. Predicted vs. Observed Mean Fitness with Building Block

shown in Figure 28 together with the observed values. This is by far the least accurate of the predictions considered here. and the inaccuracy begins in an earlier generation than that of the state or the distribution mean predictions. It is reasonable to conclude that it is the source of much of the inaccuracy in the other predictions.

5.5 Summary

This chapter develops a dynamical systems model of binary tournament selection with probabilistic thresholding. The key components of the model are an order-statistics based decision making model and a hierarchical Markov chain model (see Figure 29). Together with the probabilistic building block filtering model developed in Chapter IV, the model allows prediction of expected effectiveness resulting from a choice of exogenous parameters. The prediction of expected effectiveness serves as the basis for the parameter selection techniques proposed in Chapter VI.

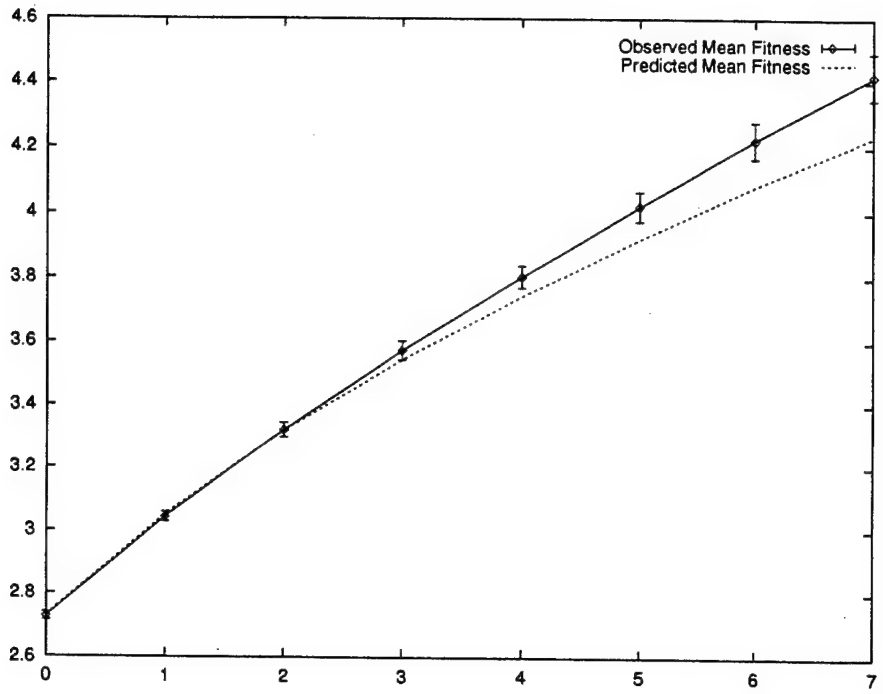


Figure 26. Predicted vs. Observed Mean Fitness without Building Block

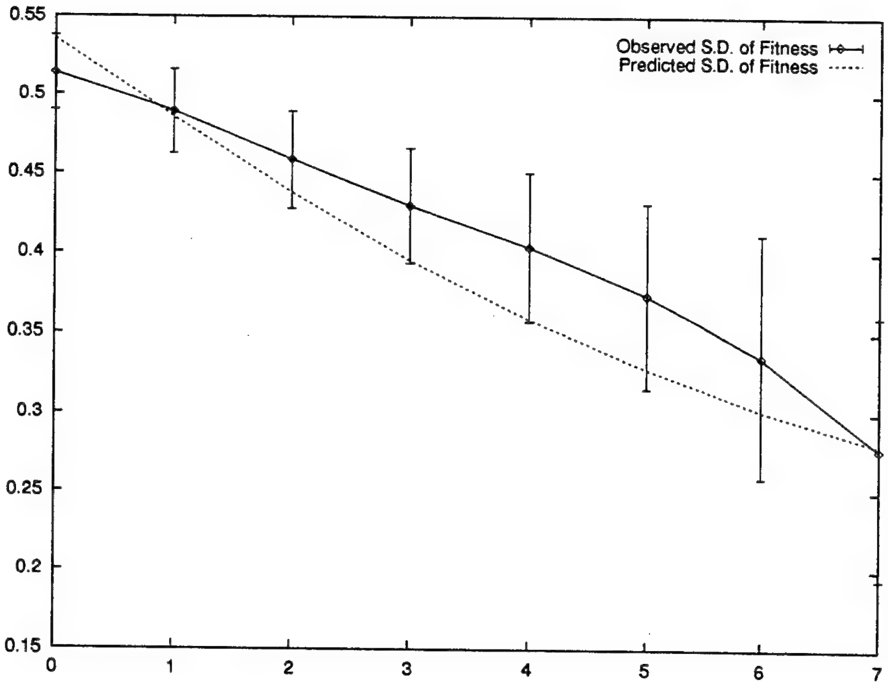


Figure 27. Predicted vs. Observed Standard Deviation of Fitness with Building Block

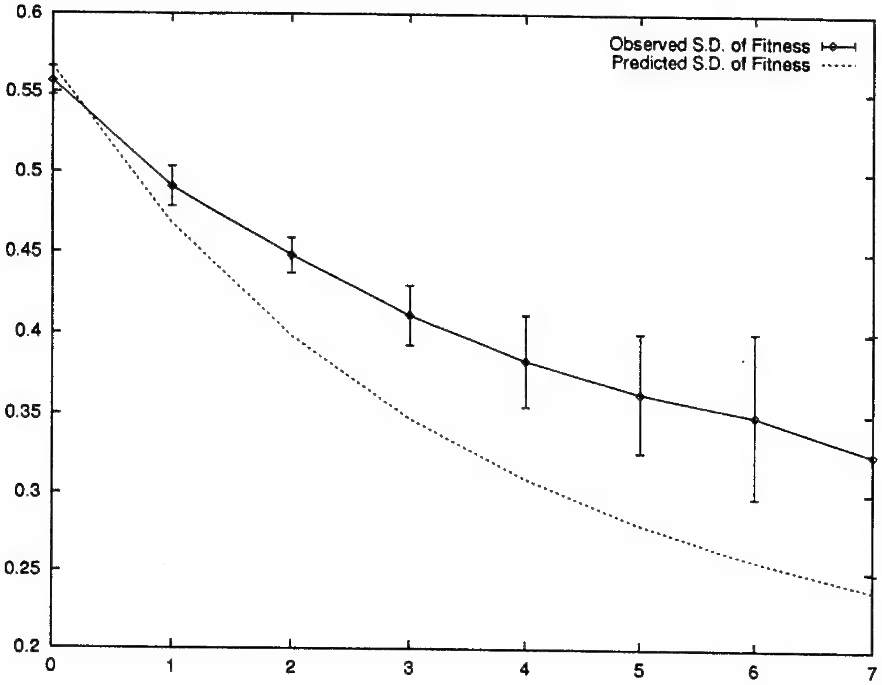


Figure 28. Predicted vs. Observed Standard Deviation of Fitness without Building Block

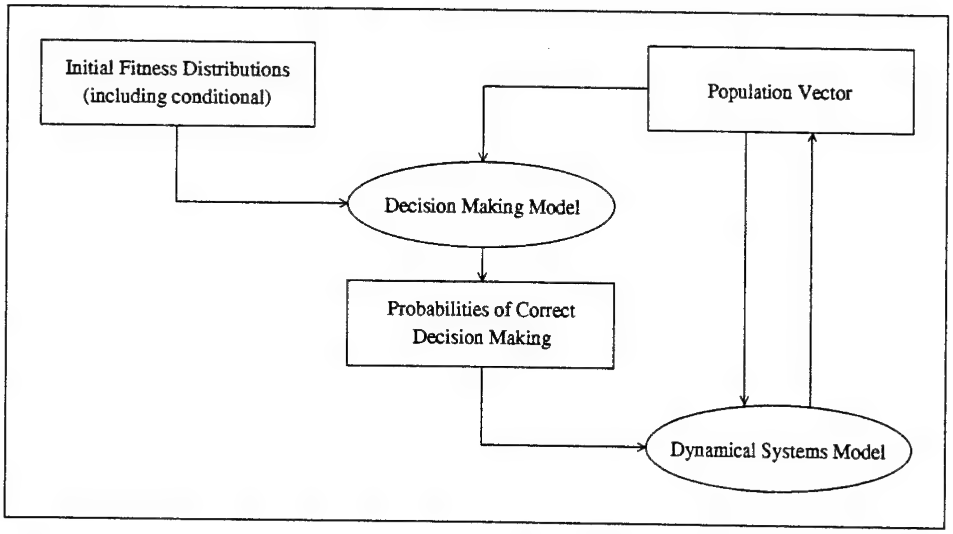

Initial Fitness Distributions (including conditional)
Population Vector
Decision Making Model
Probabilities of Correct Decision Making
Dynamical Systems Model

Figure 29. Flow of Information in Dynamical Systems Model of Binary Tournament Selection with Probabilistic Thresholding

VI. Selection of Exogenous Parameters

The fast messy genetic algorithm (fmGA) described in Section 2.6.4 is employed as an optimum seeking technique in several limited studies [27, 28, 35, 54]. These studies and this research show theoretically and empirically that the fmGA exhibits a number of advantages over the simple genetic algorithm (Section 2.4), the messy genetic algorithm (Section 2.6.3), and other optimum seeking techniques.

Practical use of the fmGA is limited by the lack of an acceptable methodology for selection of its numerous exogenous parameters, upon which its effectiveness depends. In particular, experience [28] shows that the effectiveness of the algorithm depends strongly on the filtering and thresholding parameters. Existing parameter selection techniques (see Section 2.6.5) are essentially heuristic and do not reliably yield satisfactory effectiveness in practical applications. Furthermore, they do not predict the expected effectiveness of the algorithm resulting from a given set of parameters, nor whether improved effectiveness may result from "tweaking" the parameters.

This chapter addresses the exogenous parameter selection problem for both the fmGA and the generalized fast messy genetic algorithm (gfmGA) described in Chapter III. The parameter selection problem is formally posed as an optimization problem (Section 6.1), for which the cost function is related to the expected effectiveness resulting from a particular choice of exogenous parameter settings. The definition of the cost function involves the mathematical models of probabilistic building block filtering (BBF) and binary tournament selection (BTS) with probabilistic thresholding developed in Chapters IV and V, respectively.

Because the fmGA filtering and thresholding parameters are discrete, the resulting optimization problem is combinatoric. A hill-climbing-based fmGA parameter selection technique is proposed in Section 6.2. In contrast, the gfmGA parameters are real-valued. Section 6.3 discusses the use of vector space optimization techniques to obtain a set of necessary optimality conditions (NOCs) for the parameters of the gfmGA. Parameter selection techniques for the gfmGA are proposed based on numerical solution of the NOCs and computational optimization of the cost functional.

6.1 Formal Statement of the Parameter Selection Problem

The formal statement of the linkage-friendly genetic algorithm (lfGA) exogenous parameter selection problem as an optimization problem is based on the models of probabilistic BBF and BTS with probabilistic thresholding presented in Chapters IV and V, respectively. In particular, the cost functional is defined as an error between the expected final state and the ideal final state $\hat{\mathbf{u}}$, where the expected state is defined in terms of the population vector. The population vector is of the form

$$\mathbf{p} \triangleq \begin{bmatrix} \mathbf{p}_{10} & \mathbf{p}_{11} & \cdots & \mathbf{p}_{1\ell} \\ \mathbf{p}_{20} & \mathbf{p}_{21} & \cdots & \mathbf{p}_{2\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{p}_{m0} & \mathbf{p}_{m1} & \cdots & \mathbf{p}_{m\ell} \end{bmatrix} .$$

Each component $\mathbf{p_{ij}}$ is of the form of Equation 12, where A is the class of individuals containing building block i. The $\mathbf{p_{ij}}$'s are viewed as conditional probabilities that an individual randomly drawn from the population contains building block i given that it is of length j.

The matrix $\underline{\mathbf{u}}(\mathbf{x}, t)$ is defined such that the i, jth component

$$u_{ij}(\mathbf{x}, t) \triangleq \psi(i; t)(\mathbf{1}^\top \mid \mathbf{0}^\top)\mathbf{p_{ij}}\mathbf{1}$$

is the probability that an individual randomly sampled from population $P(t)$ contains building block i and is of length j, given the exogenous parameter set $\mathbf{x}$. The *expected state* in generation t is the vector

$$\underline{\mathbf{J}}_{\mathbf{u}}(\mathbf{x}, t) \triangleq \underline{\mathbf{u}}(\mathbf{x}, t)\mathbf{1} ,$$

the ith component of which is the probability that an individual randomly sampled from population $P(t)$ contains building block i, given the exogenous parameter set $\mathbf{x}$.

The exogenous parameters (excluding the $\psi(0; t, k)$'s and $\theta(0: i, j; k, t$'s) for iteration k of the generalized fast messy genetic algorithm are taken from the vector space $\mathbb{X} \triangleq \mathbb{R}^{\ell \times t_p} \times \mathbb{R}^{\ell \times (\ell+1) \times (\ell+1) \times t_p}$. They are subject to the inequality constraints given in Equations 6 and 7, which are represented formally by

$$G(\mathbf{x}) \triangleq \begin{bmatrix} \hat{G}(\mathbf{x}) \\ -\mathbf{x} \end{bmatrix} \leq \mathbf{0_Z} \ .$$

where $G : \mathbb{X} \longrightarrow Z$.

The exogenous parameter selection problem may thus be formally stated as:

Find the set of exogenous parameters $\mathbf{x} \in \mathbb{X}$ which minimize the cost functional $J : \mathbb{X} \longrightarrow \mathbb{R}$,

$$J(\mathbf{x}) \triangleq \frac{1}{2} ||\underline{\mathbf{J}}_{\mathbf{u}}(\mathbf{x}) - \hat{\mathbf{u}}||_2^2 \ .$$

subject to $G(\mathbf{x}) \leq \mathbf{0_Z}$.

6.2 fmGA Parameter Selection

This section proposes an exogenous parameter selection technique for the fast messy genetic algorithm which bears some resemblance to the technique proposed by Kargupta. The most significant advantage of this technique over Kargupta's is that the choice of thresholding parameters explicitly considers the expected effectiveness of the algorithm. Also, as a consequence of the underlying tournament selection model (Chapter V), this choice reflects the dynamic nature of the probability of correct decision making. Another advantage is that all of the design parameters required by the technique (the nominal string length ℓ, the estimated level of deception k, and the assumed initial fitness distributions) are already required by the fast messy genetic algorithm.

Luenberger presents numerous mathematical techniques for the optimizatioon of functionals defined on vector spaces or subsets of vector spaces satisfying specific conditions [50]. The remainder of this section considers their use in the fmGA parameter selection problem, and shows that they are not, in general, directly applicable.

Both the pre-Hilbert space form and the classical form of the Projection Theorem require that the set over which optimization is performed be a vector space. The same is true of the techniques presented for solution of minimum norm problems. When $\ell + 1$ is not prime, no operations $\oplus$ and $\otimes$ exist such that $\mathcal{B}$ together with $\oplus$ and $\otimes$ form a vector space. Thus, the projection theorems and the minimum norm problem techniques are not, in general, directly applicable.

The Fenchel Duality Theorem and the Lagrange Multiplier theorems for global theory require that the set over which optimization is performed be a convex subset of a vector space. Again, unless $\ell + 1$ is prime, $\mathcal{B}$ does not satisfy this condition, so these theorems are also not, in general, directly applicable.

The Lagrange Multiplier theorem for local theory requires that the functional to be optimized be Fréchet-differentiable, which implies that the set on which it is defined is a vector space. Likewise, the Generalized Kuhn-Tucker Theorem requires that the functional to be optimized be defined on a vector space. Thus, these theorems are also not, in general, directly applicable.

The proposed technique is as shown in Figure 30. A possible disadvantage of this technique is the computationally intensive nature of the second step, in which the optimal selection episode duration and associated effectiveness are determined for each meaningful choice of the threshold parameter. For "difficult"

1. Take $\lambda^{(0)} = \ell - k$. Set $e = 0$.
2. For each candidate threshold $\hat{\theta} \in \{2\lambda^{(e)} - \ell, \ldots, \lambda^{(e)}\}$, find $\hat{t}$ which minimizes J.
3. Take $\theta^{(e)}$ to be the $\hat{\theta}$ which yields the overall minimum J. Take t_e^* to be the corresponding $\hat{t}$.
4. Take
$$\lambda^{(e+1)} = \min \left\{ \lambda : \eta_{e+1} \min_i \frac{q_{i,t_e^*}}{q_{i,0}} \geq 1 \right\}$$
5. Set $e = e + 1$. If $\lambda^{(e)} > k$ goto step 2.

Figure 30. Fast Messy Genetic Algorithm Parameter Selection Technique

optimization problems, multiple independent runs of the fast messy genetic algorithm are necessary, and it is appropriate to "amortize" the computational cost of parameter selection over the number of runs performed. Also, the same (or better) overall effectiveness may result from a smaller number of runs of a more effective algorithm as from a larger number of runs of a less effective algorithm. Thus, for "difficult" optimization problems, the computational cost of the second step is justified.

The technique may be viewed as a hillclimbing strategy, in the sense that each instance of the third step specifies a locally optimal choice of the threshold parameter and selection episode duration. Kargupta's technique may also be viewed as a hillclimbing strategy, with a different criterion for local optimality which does not consider expected effectiveness.

The choice of filtering parameters in the fourth step is motivated by the stated design objective of Goldberg, et al. [35] and Kargupta [47]. That is, it ensures that after filtering each building block is expected to have at least one copy in the population. Importantly, the choice is made based on the expected number of copies of the least well represented building block, and the model does not assume that the number of copies of that building block doubles in each generation of tournament selection.

6.3 gfmGA Parameter Selection

The filtering and thresholding parameters of the gfmGA are real-valued. Furthermore, the cost function defined in Section 6.1 is continuously differentiable with respect to the parameters. Consequently, vector space optimization techniques [50] may be used to obtain necessary optimality conditions (NOCs) for the parameter selection problem. This section discusses the application of the Generalized Kuhn-Tucker Theorem to obtain NOCs, and discusses parameter selection techniques for the gfmGA. Luenberger states the Generalized Kuhn-Tucker Theorem essentially as follows:

Theorem 6.3.1 (Generalized Kuhn-Tucker) *Let X be a vector space and Z a normed space having positive cone P. Assume that P contains an interior point. Let f be a Gateaux differentiable*[1] *real-valued functional on X and G a Gateaux differentiable mapping from X into Z. Assume that the Gateaux differentials are linear in their increments. Suppose x_0 minimizes f subject to $G(x) \leq 0_Z$ and that x_0 is a regular point of the inequality $G(x) \leq 0_Z$. Then there is a $z_0^* \in Z^*$, $z_0^* \geq 0_{Z^*}$ such that the Lagrangian*

$$f(x) + \langle G(x), z_0^* \rangle$$

is stationary at x_0; furthermore, $\langle G(x_0), z_0^* \rangle = 0$.

Proof: See Luenberger [50]. ■

As an immediate consequence of this theorem, a set of necessary optimality conditions for the exogenous parameter selection problem is obtained.

Corollary 6.3.2 *Let the cost function J and the constraint mapping G be as defined in Section 6.1. Suppose $\mathbf{x_0}$ minimizes J subject to $G(\mathbf{x}) \leq 0_Z$. Then there exist $z_m \in \mathbb{R}^m$, $z_n \in \mathbb{R}^n$, $z_m \geq 0_{\mathbb{R}^m}$, $z_n \geq 0_{\mathbb{R}^n}$ such that*

$$\begin{aligned} z_m &\geq 0_{\mathbb{R}^m} \ . & (29) \\ z_n &\geq 0_{\mathbb{R}^n} \ . & (30) \\ J_x(\mathbf{x_0}) + z_m^\top \hat{G}_x(\mathbf{x_0}) - z_n^\top &= 0_{\mathbb{R}^n} \ . & (31) \end{aligned}$$

and

$$z_m^\top \hat{G}(\mathbf{x_0}) = z_n^\top \mathbf{x_0} = 0 \tag{32}$$

[1] The Gateaux differential is the generalization to arbitrary vector spaces of the directional differential. Following Luenberger [50], let X be a vector space, Y a normed space, $D \subseteq X$, $T : D \longrightarrow Y$, $x \in D$, and $h \in X$. If the limit

$$\delta T(x; h) \triangleq \lim_{\alpha \to 0} \frac{1}{\alpha}[T(x + \alpha h) - T(x)]$$

exists, it is called the *Gateaux differential of T at x with increment h*. If the limit exists for each $h \in X$, the transformation T is said to be *Gateaux differentiable at x*.

Proof: The constraint space $Z = \mathbb{R}^{m+n}$ is a Euclidean vector space, hence the positive cone P is the first orthant. P contains interior points, e.g. $(1, \ldots, 1)$. The transition operators τ_m and τ_s defined in Chapters IV and V, respectively, are both differentiable with respect to each of the filtering and thresholding parameters. Consequently, the mapping $\underline{\mathbf{J}}_{\mathbf{u}}$ defined in Section 6.1 is also differentiable with respect to the parameters, and furthermore, so is J. The constraint mapping G is also differentiable with respect to the parameters, and every point satisfying $G(\mathbf{x}) \leq 0_Z$ also is a regular point of the inequality (i.e. there are no cusps in the constraint boundaries). Thus, the conditions of Theorem 6.3.1 are satisfied.

Up to isomorphism, $J : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $G : \mathbb{R}^n \longrightarrow \mathbb{R}^{m+n}$. Suppose $x_0 \in \mathbb{R}^n$ minimizes J subject to $G(x_0) \leq 0_{\mathbb{R}^{m+n}}$. Then x_0 is a regular point of $G(x) \leq 0_{\mathbb{R}^{m+n}}$, and (following Luenberger [50:Ex. 2, §9.4]) the constraint may be written

$$\hat{G}(x_0) \leq 0_{\mathbb{R}^m} \tag{33}$$

and

$$-x_0 \leq 0_{\mathbb{R}^n} \quad . \tag{34}$$

where $\hat{G} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$.

Theorem 6.3.1 implies that there exist $z_m \in \mathbb{R}^m$ and $z_n \in \mathbb{R}^n$ such that

$$z_m \geq 0_{\mathbb{R}^m} \quad , \tag{35}$$

$$z_n \geq 0_{\mathbb{R}^n} \quad , \tag{36}$$

$$J(x) + z_m^{\top}\hat{G}(x) + z_n^{\top}(-x) \qquad \text{is stationary at } x_0 \quad , \tag{37}$$

and

$$z_m^{\top}\hat{G}(x_0) + z_n^{\top}(-x_0) = 0 \quad . \tag{38}$$

Condition 37 may be written

$$J_x(x_0) + z_m^{\top}\hat{G}_x(x_0) - z_n^{\top} = 0$$

Conditions 33 and 35 imply that $z_m^\top \hat{G}(x_0) \leq 0$, while Conditions 34 and 36 imply that $-z_n^\top x_0 \leq 0$. These conditions, together with Condition 38 imply that $z_m^\top \hat{G}(x_0) = z_n^\top x_0 = 0$, which completes the proof. ■

Equations 29 through 32, which form a system of simultaneous non-linear equations in the exogenous parameters, are referred to as the necessary optimality conditions (NOCs). Because the cost functional J is continuous on the feasible region, which is a compact subset of a metric space, J attains its minimum on the region (see Theorem 4.28 of Apostol [3]), i.e. there exists an $\mathbf{x_0}$ which minimizes J subject to $G(\mathbf{x}) \leq \mathbf{0_Z}$. The parameter set $\mathbf{x_0}$ yields optimal expected effectiveness of the generalized fast messy genetic algorithm. In principle, the exogenous parameter selection problem reduces to the problem of finding $\mathbf{x_0}$.

By the preceding argument, the existence of at least one solution $\mathbf{x_0}$ of the NOCs is guaranteed. Under certain conditions, the solution is unique, in which case Equations 29 through 32 are both necessary and sufficient for optimality. In particular, if the Hessian matrix[2] of J is positive definite on the entire feasible region, then there exists a unique minimum of J on the region, and hence $\mathbf{x_0}$ is unique. Because the constraints $G(\mathbf{x}) \leq \mathbf{0_Z}$ define a convex region of the parameter space, $\mathbf{x_0}$ is also unique in the more general case that J is convex on the region. Finally, $\mathbf{x_0}$ may be unique even if J is not convex.

Analysis of the positive definiteness of the Hessian matrix of J via explicit derivation of the partial derivatives is tedious and unrewarding, as is explicit analysis of the convexity of J. The question of the uniqueness of $\mathbf{x_0}$ may be addressed more directly. Because $\mathbf{J_a}$ is well approximated by a high-order polynomial in the exogenous parameters, the roots of which depend on the objective function, it seems likely that there exist (many) objective functions for which the stationary points of J include points of local maximum, saddle points, and multiple points of local minimum. For generality, it is assumed in the sequel that the solution of the NOCs is not unique.

[2] Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $p \in \mathbb{R}^n$. Then the matrix A whose components are

$$A_{ij} \triangleq \left. \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right|_p$$

is the *Hessian matrix of f at p*[64]

In principle, the parameter selection problem reduces to the problem of finding the solution $\mathbf{x_0}$ of the NOCs for which J is minimized. In practice, the explicit form of the resulting expressions does not suggest a direct solution technique, despite extensive analysis and consultation. Thus, solutions must be obtained numerically. Standard techniques for simultaneous solution of non-linear equations include Newton-Raphson, globally convergent extensions thereof, and Broyden's Method [64].

Newton-Raphson is perhaps the simplest and best known multidimensional root finding technique. Given a "good" initial guess of the location of a root, it converges quadratically to the root. For a system of equations of the form $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ with Jacobian matrix $\mathbf{J}$, the update rule is

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \delta\mathbf{x} \quad ,$$

where $\delta\mathbf{x}$ satisfies

$$\mathbf{J} \cdot \delta\mathbf{x} = -\mathbf{F} \quad .$$

Given a "poor" initial guess, Newton-Raphson fails to converge. Variations of the algorithm overcome this significant limitation by requiring that each step reduce $f \triangleq \frac{1}{2}|\mathbf{F}|^2$. This is possible because each step is in a *descent direction* for f. Thus, either the full step decreases f, or a smaller step in the same direction can be found which decreases f. Both Newton-Raphson and its globally convergent extensions require the existence of the Jacobian matrix. This condition is satisfied by the NOCs, so that these techniques are applicable.

Even though the Jacobian matrix exists and its analytical form is available, its evaluation is computationally intensive. Consequently, multidimensional secant methods, such as Broyden's method, may be more efficient than Newton-Raphson. A thorough discussion of this technique, as well as a reference to the primary literature, may be found in Press, et al. [64].

The exogenous parameter selection problem for generalized fast messy genetic algorithms may be approached by identifying solutions to the NOCs and selecting the solution for which the cost functional J

is minimized. A parameter selection technique for generalized fast messy genetic algorithms may be stated as shown in Figure 31. This technique satisfies the formal acceptability criteria established in Chapter I.

```
1. Let $\mathbf{x_0}$ be the best currently known set of exogenous parameters for the fast
   messy genetic algorithm.
2. While the termination condition is not satisfied:
   (a) Obtain a solution $\mathbf{x}$ of the NOCs.
   (b) If $J(\mathbf{x}) < J(\mathbf{x_0})$ then replace $\mathbf{x_0}$ by $\mathbf{x}$.
```

Figure 31. Generalized Fast Messy Genetic Algorithm Parameter Selection Technique Based on Solution of the Necessary Optimality Conditions

The criteria are:

1. the technique guarantees expected effectiveness no worse than that resulting from the best set of parameters obtained using existing techniques.
2. the technique requires no *a priori* knowledge of the optimal solution.
3. the technique requires no design parameters beyond those of the linkage-friendly genetic algorithm; and
4. the computational effort required by the technique scales well with the effort required by the linkage-friendly genetic algorithm.

The technique is essentially a simplistic search algorithm. It generates candidates from the set of solutions of the NOCs, which includes the local maxima and saddle points of J, as well as the local minima. The amount of computation required to obtain a solution to the NOCs is approximately that required to obtain a local minimum of the cost function J. Thus. the technique is likely to be less efficient than one which randomly generates candidates from the set of local minima (see Figure 32).

Even more promising strategies result from the use of standard constrained optimum seeking techniques to minimize J. The literature abounds with applicable techniques, including simulated annealing, tabu search, and evolutionary algorithms. Because the cost function is continuously differentiable, it is worthwhile

```
1. Let $\mathbf{x_0}$ be the best currently known set of exogenous parameters for the fast messy genetic algorithm.
2. While the termination condition is not satisfied:
    (a) Obtain a point of local minimum $\mathbf{x}$ for $J$.
    (b) If $J(\mathbf{x}) < J(\mathbf{x_0})$ then replace $\mathbf{x_0}$ by $\mathbf{x}$.
```

Figure 32. Generalized Fast Messy Genetic Algorithm Parameter Selection Technique Based on Optimization of the Cost Function

to consider hybrid techniques which combine a globally convergent technique (e.g. genetic algorithms) with an efficient local optimization technique (e.g. conjugate gradient). Such hybrids serve as effective optimum seeking techniques for other objective functions with similar properties [55, 56].

6.4 Summary

The linkage-friendly genetic algorithm exogenous parameter selection problem is formally posed as a constrained optimization problem. By viewing the fast messy genetic algorithms parameter selection problem in this way, a hillclimbing technique is obtained which represents a substantial improvement over existing techniques. The Generalized Kuhn-Tucker Theorem is employed to obtain necessary optimality conditions (NOCs) for the generalized fast messy genetic algorithm parameter selection problem. Several techniques are suggested by which the problem may be solved in practice, including numerical solution of the NOCs and a hybrid genetic algorithm which incorporates efficient local optimization.

VII. Conclusions and Recommendations

The primary objectives of this research are to

- mathematically model those properties of specific linkage-friendly genetic algorithms which are related to expected effectiveness; and
- develop exogenous parameter selection techniques for those linkage-friendly genetic algorithms, focusing on maximizing their expected effectiveness.

The major conclusions are summarized in Section 7.1. and recommendations for future research are presented in Section 7.2

7.1 Conclusions

Formal framework for evolutionary algorithms. Evolutionary algorithms are a class of stochastic population-based algorithms which are commonly applied as optimum seeking techniques. A novel framework for evolutionary algorithms is proposed which extends the work of Bäck and Schwefel (Section 2.3). Within this formal framework, evolutionary operators are viewed as mappings from parameter spaces to random population transformations. Definitions of recombination, mutation, and selection operators are proposed which capture their distinguishing characteristics.

Linkage-friendly genetic algorithms (lfGAs). The class of lfGAs consists of evolutionary algorithms which use order-invariant representation schemes and strictly invariant selection operators. Previously studied examples of the class include the messy genetic algorithm (mGA) and the fast messy genetic algorithm (fmGA), which are defined within the formal framework for evolutionary algorithms in Sections 2.6.3 and 2.6.4, respectively.

The mGA and fmGA represent theoretical steps towards effective linkage-friendly genetic algorithms. However, the mGA is $O([\text{card}(\mathcal{A}) \cdot \ell]^k)$ in time and space, where $\mathcal{A}$ is the genic alphabet, ℓ is the problem size, and k is the building block size. The fmGA addresses this drawback, but it also introduces numerous exogenous parameters for which no practical selection methodology is known. Experience shows that the effectiveness of the fmGA is highly sensitive to the choice of these exogenous parameters [28].

Chapter III proposes a novel lfGA, the generalized fast messy genetic algorithm (gfmGA), which uses probabilistic generalizations of the filtering (mutation) and selection operators used by the fmGA. The fmGA is a special case of the gfmGA. Consequently, existence is guaranteed of parameters for which the gfmGA expected effectiveness is no worse than the best possible fmGA expected effectiveness (Section 3.3).

Dynamical systems models of probabilistic operators. The practical application of the fmGA is limited by the lack of an acceptable parameter selection methodology. Existing techniques are handicapped by a poor understanding of the relationship between the filtering and thresholding parameters of the algorithm and the expected effectiveness. This research develops a dynamical systems model of the gfmGA (and of the fmGA as a special case) which predicts the expected effectiveness as a function of the filtering and thresholding parameters. The key elements of the model are:

1. **Probability of building block presence after probabilistic filtering.** Previous models of building block filtering considered only deterministic and destructive filtering. This research (Chapter IV) extends these models to consider probabilistic and possibly increasing individual lengths. Probabilities of survival and construction are combined to yield the total probability of building block presence following filtering.

2. **Order statistical analysis of the probability of correct decision making.** Early linkage-friendly genetic algorithm studies aim at improving probabilities of correct decision making (whether or not this is explicitly stated), but those probabilities are inadequately modeled. Previous models of tournament selection focus on either takeover time or selection intensity. Neither model provides information regarding the relative growth of one class of individuals with respect to another (except the growth of the "best" individuals with respect to the "worst" individuals). This research develops the probability of correct decision making exactly and explicitly in terms of the initial fitness distributions of the

competing classes and the number of ancestors belonging to each of the classes for each competitor (Section 5.1).

3. **Markov chain analysis of competition in the presence of non-trivial thresholding.** Another limitation of previous tournament selection models is that they neglect thresholding. By affecting the pairs of individuals which are considered compatible, thresholding affects not only the effective probability of correct decision making, but also the effective selective pressure. This research uses Markov chain analysis to develop an exact dynamical systems model of competing classes of individuals under binary tournament selection with (probabilistic) thresholding (Section 5.2).

Parameter selection techniques based on maximizing expected effectiveness. The mathematical model developed permits the design of parameter selection techniques which explicitly consider the expected effectiveness of the algorithm. This research considers a parameter selection technique to be acceptable if it satisfies the following criteria:

1. the technique guarantees expected effectiveness no worse than that resulting from the best set of parameters obtained using existing techniques,
2. the technique requires no *a priori* knowledge of the optimal solution.
3. the technique requires no design parameters beyond those of the linkage-friendly genetic algorithm; and
4. the computational effort required by the technique scales well with the effort required by the linkage-friendly genetic algorithm.

The parameter selection problem is formally posed as a constrained optimization problem (Section 6.1). An fmGA parameter selection technique based on hill-climbing is proposed which satisfies the acceptability criteria (Section 6.2). In part because the gfmGA parameters are real-valued, vector space optimization techniques (specifically. the Generalized Kuhn-Tucker Theorem) may be used to obtain formal necessary optimality conditions (NOCs) for the gfmGA parameters (Section 6.3). One gfmGA parameter selection technique is proposed which is based on numerical solution of the NOCs. A second technique is proposed based on computational optimization of the cost functional.

7.2 Recommendations

This research answers a number of questions regarding the properties of linkage-friendly genetic algorithms. It also suggests a number of promising areas for additional research:

1. **Fitness distributions after building block filtering.** The building block filtering model developed in Chapter IV considers only the probability of building block presence after filtering. It provides no information regarding the resulting fitness distributions. The availability of such information would provide the initial fitness distributions required to model the tournament selection episode following the filtering event.

2. **Non-monotonicity of the probability of correct decision making.** The use of thresholding in binary tournament selection is predicated on the assumption that the probability p_d of correct decision making depends on the thresholding metric. In particular, the messy genetic algorithm and fast messy genetic algorithm implicitly assume that p_d is a non-decreasing function of the number of common defining loci. Limited empirical results (not reported here) based on the tournament selection model developed in Chapter V suggest that this assumption is incorrect. These results suggest that better decision making may result from a compatibility criteria which places both upper and lower bounds on the number of common defining loci.

3. **Extension of tournament selection model to competition between N classes.** The mathematical model of tournament selection developed in this research (Sections 5.1 and 5.2) focuses on competition between two classes of individuals. It is natural to extend the model to competition between N classes of individuals. Such an extension would, for example. facilitate more accurate modeling of the effects of the presence of individuals which contain multiple building blocks.

4. **Efficiency considerations of parameter selection.** This research considered only effectiveness in the definition of algorithm performance. Another important aspect of performance is efficiency. Future research should examine the appropriate definition of an efficiency functional, and the appropriate means by which to consider both effectiveness and efficiency in selecting exogenous parameters. Performance may be defined as a convex combination of effectiveness and efficiency. Alternatively, the parameter selection problem may be viewed as a multi-objective optimization problem.

5. **Application of the gfmGA to practical problems.** Future research also includes application of the gfmGA to real world problems. such as the polypeptide structure prediction problem. The AFIT/WL Genetic Computation Techniques (AGCT) research group performs a number of state-of-the-art investigations in the application of evolutionary algorithms to this problem (e.g. [54]). Several issues must be addressed.

    - The inherently discrete nature of gfmGA individual spaces strongly suggests that reasonable effectiveness may be expected only for objective functions with a combinatoric character. The polypeptide structure prediction problem exhibits both combinatoric and continuous characteristics. which suggests hybridization of the gfmGA with efficient local minimization techniques (c.f. [56]).

    - The computational resources necessary to solve a real-world polypeptide structure prediction problem require the use of high-performance scalable architectures. Existing mappings of the fmGA to such architectures (e.g. [28]) provide a reasonable point of departure for determining appropriate mappings of the gfmGA. Appropriate mappings of the parameter selection techniques to scalable architectures are also required.

    - The prediction of expected effectiveness. and consequently the selection of gfmGA parameters. requires estimation of the initial fitness distributions. This estimate may be obtained by assumptions based on physical insight (e.g. distributional form. signal difference), and parameter estimates based on a uniform sampling of conformation space (e.g. mean, variance).

Appendix A. Cardinalities for Decision Making Model

The distributions of fitnesses in a uniform random population developed in Section 5.3 are expressed in terms of certain cardinalities of subsets of I and I^2, where I is the individual space. This appendix presents expressions for these cardinalities. $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$.

$$\begin{aligned}
N &\triangleq \operatorname{card}\left(\{(a.b) \in I(\lambda) \times I(\lambda) : \Lambda_c(a.b) = \lambda_c\}\right) \\
&= |\mathcal{A}|^{\lambda} \binom{\ell}{\lambda} \cdot |\mathcal{A}|^{\lambda} \binom{\lambda}{\lambda_c} \binom{\ell - \lambda}{\lambda - \lambda_c} \\
N_{\beta} &\triangleq \operatorname{card}\left(\{(a.b) \in I_{\beta}(\lambda) \times I(\lambda) : \Lambda_c(a.b) = \lambda_c\}\right) \\
&= |\mathcal{A}|^{\lambda-k} \binom{\ell - k}{\lambda - k} \cdot |\mathcal{A}|^{\lambda} \binom{\lambda}{\lambda_c} \binom{\ell - \lambda}{\lambda - \lambda_c} \\
N_{\beta,\beta} &\triangleq \operatorname{card}\left(\{(a.b) \in I_{\beta}(\lambda) \times I_{\beta}(\lambda) : \Lambda_c(a,b) = \lambda_c\}\right) \\
&= |\mathcal{A}|^{\lambda-k} \binom{\ell - k}{\lambda - k} \cdot |\mathcal{A}|^{\lambda-k} \binom{\lambda - k}{\lambda_c - k} \binom{\ell - \lambda}{\lambda - \lambda_c} \\
N_{\beta,\neg\chi_i} &\triangleq \operatorname{card}\left(\{(a.b) \in I_{\beta}(\lambda) \times I_{\neg\chi_i}(\lambda) : \Lambda_c(a,b) = \lambda_c\}\right) \\
&= \begin{cases}
|\mathcal{A}|^{\lambda-k} \tbinom{\ell-k}{\lambda-k} \cdot \sum_{r=0}^{k} |\mathcal{A}|^{\lambda-r} \tbinom{k}{r} \tbinom{\lambda-k}{\lambda_c-r} \tbinom{\ell-\lambda}{\lambda-\lambda_c}, & \text{if } i = \beta \\[4ex]
|\mathcal{A}|^{\lambda-k} \sum_{r_{ai=0}}^{k} \Big[\tbinom{k}{r_{ai}} \tbinom{\ell-k}{\lambda-k-r_{ai}} & \\
\quad \cdot \sum_{r_{bi}=0}^{k} |\mathcal{A}|^{\lambda-r_{bi}} \sum_{s_i=0}^{r_{ai}} \tbinom{r_{ai}}{s_i} \tbinom{k-r_{ai}}{r_{bi}-s_i} \tbinom{\lambda-r_{ai}}{\lambda_c-s_i} \tbinom{\ell-\lambda-(k-r_{ai})}{\lambda-\lambda_c-(r_{bi}-s_i)} \Big]. & \text{if } i \neq \beta
\end{cases} \\
N_{\beta,\beta\neg\chi_i} &\triangleq \operatorname{card}\left(\{(a,b) \in I_{\beta}(\lambda) \times (I_{\beta}(\lambda) \cap I_{\neg\chi_i}(\lambda)) : \Lambda_c(a.b) = \lambda_c\}\right) \\
&= \begin{cases}
0, & \text{if } i = \beta \\[4ex]
|\mathcal{A}|^{\lambda-k} \sum_{r_{ai}=0}^{k} \Big[\tbinom{k}{r_{ai}} \tbinom{\ell-k}{\lambda-k-r_{ai}} & \\
\quad \cdot \sum_{r_{bi}=0}^{k} |\mathcal{A}|^{\lambda-k-r_{bi}} \sum_{s_i=0}^{r_{ai}} \tbinom{r_{ai}}{s_i} \tbinom{k-r_{ai}}{r_{bi}-s_i} \tbinom{\lambda-k-r_{ai}}{\lambda_c-k-s_i} \tbinom{\ell-\lambda-(k-r_{ai})}{\lambda-\lambda_c-(r_{bi}-s_i)} \Big], & \text{if } i \neq \beta
\end{cases} \\
N_{\beta\neg\chi_i} &\triangleq \operatorname{card}\left(\{(a.b) \in (I_{\beta}(\lambda) \cap I_{\neg\chi_i}(\lambda)) \times I(\lambda) : \Lambda_c(a,b) = \lambda_c\}\right) \\
&= \begin{cases}
0, & \text{if } i = \beta \\
\sum_{r_{ai}=0}^{k} |\mathcal{A}|^{\lambda-k-r_{ai}} \tbinom{k}{r_{ai}} \tbinom{\ell-k}{\lambda-k-r_{ai}} \cdot |\mathcal{A}|^{\lambda} \tbinom{\lambda}{\lambda_c} \tbinom{\ell-\lambda}{\lambda-\lambda_c}, & \text{if } i \neq \beta
\end{cases}
\end{aligned}$$

$$
\begin{aligned}
N_{\beta\neg\chi_i,\beta} &\triangleq \operatorname{card}\left(\{(a.b) \in (I_\beta(\lambda) \cap I_{\neg\chi_i}(\lambda)) \times I\beta(\lambda) : \Lambda_c(a.b) = \lambda_c\}\right) \\
&= \begin{cases} 0, & \text{if } i = \beta \\ \sum_{r_{ai}=0}^{k} |\mathcal{A}|^{\lambda-k-r_{ai}} \binom{k}{r_{ai}} \binom{\ell-k}{\lambda-k-r_{ai}} \cdot |\mathcal{A}|^{\lambda-k\cdot} \binom{\lambda-k}{\lambda_c-k} \binom{\ell-\lambda}{\lambda-\lambda_c}, & \text{if } i \neq \beta \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
N_{\beta\neg\chi_i,\neg\chi_i} &\triangleq \operatorname{card}\left(\{(a.b) \in (I_\beta(\lambda) \cap I_{\neg\chi_i}(\lambda)) \times I_{\neg\chi_i}(\lambda) : \Lambda_c(a.b) = \lambda_c\}\right) \\
&= \begin{cases} 0, & \text{if } i = \beta \\ \sum_{r_{ai}=0}^{k} \Big[|\mathcal{A}|^{\lambda-k-r_{ai}} \binom{k}{r_{ai}} \binom{\ell-k}{\lambda-k-r_{ai}} & \\ \cdot \sum_{r_{bi}=0}^{k} |\mathcal{A}|^{\lambda-r_{bi}} \sum_{s_i=0}^{r_{ai}} \binom{r_{ai}}{s_i} \binom{k-r_{ai}}{r_{bi}-s_i} \binom{\lambda-r_{ai}}{\lambda_c-s_i} \binom{\ell-\lambda-(k-r_{ai})}{\lambda-\lambda_c-(r_{bi}-s_i)} \Big], & \text{if } i \neq \beta \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
N_{\beta\neg\chi_i.\beta\neg\chi_i} &\triangleq \operatorname{card}\left(\{(a.b) \in (I_\beta(\lambda) \cap I_{\neg\chi_i}(\lambda)) \times (I_\beta(\lambda) \cap I_{\neg\chi_i}(\lambda)) : \Lambda_c(a.b) = \lambda_c\}\right) \\
&= \begin{cases} 0, & \text{if } i = \beta \\ \sum_{r_{ai}=0}^{k} \Big[|\mathcal{A}|^{\lambda-k-r_{ai}} \binom{k}{r_{ai}} \binom{\ell-k}{\lambda-k-r_{ai}} & \\ \cdot \sum_{r_{bi}=0}^{k} |\mathcal{A}|^{\lambda-k-r_{bi}} \sum_{s_i=0}^{r_{ai}} \binom{r_{ai}}{s_i} \binom{k-r_{ai}}{r_{bi}-s_i} \binom{\lambda-k-r_{ai}}{\lambda_c-s_i} \binom{\ell-\lambda-(k-r_{ai})}{\lambda-\lambda_c-(r_{bi}-s_i)} \Big], & \text{if } i \neq \beta \end{cases}
\end{aligned}
$$

Appendix B. Numerical Techniques Used in Tournament Selection Experiments

The computational experiments reported in Chapter V assume that the unconditional fitness densities f and g and the conditional fitness densities f_Ω and g_Ω of the competing classes are those of the normal distributions $N(\mu_A, \sigma_A^2)$, $N(\mu_B, \sigma_B^2)$, $N(\mu_{A|\Omega}, \sigma_{A|\Omega}^2)$ and $N(\mu_{B|\Omega}, \sigma_{B|\Omega}^2)$, respectively. Consequently, the probabilities of correct decision making for individuals with more than one ancestor do not have closed form solutions. This appendix discusses the numerical techniques used to compute the probabilities.

The integral

$$I \;\triangleq\; \int_{-\infty}^{\infty} f_\Omega(t)[F(t)]^{n_X^{(a)}-1}[G(t)]^{n_Y^{(a)}} \int_{-\infty}^{t} g_\Omega(s)[F(s)]^{n_X^{(b)}}[G(s)]^{n_Y^{(b)}-1}\, ds\, dt$$

may be formally expressed as

$$I \;=\; \int_{-\infty}^{\infty} f_\Omega(t)[F(t)]^{n_X^{(a)}-1}[G(t)]^{n_Y^{(a)}} I_1(-\infty, t)\, dt \quad ,$$

where

$$I_1(t_0, t_f) \;\triangleq\; \int_{t_0}^{t_f} g_\Omega(s)[F(s)]^{n_X^{(b)}}[G(s)]^{n_Y^{(b)}-1}\, ds \quad .$$

Because the integral operator is additive with respect to the interval of integration.

$$I_1(-\infty, x) \;=\; I_1(-\infty, a_0) + \sum_{i=1}^{N} I_1(a_{i-1}, a_i) + I_1(a_N, x) \quad .$$

Each of the integrals is evaluated numerically, using Romberg integration [64]. The first integral is improper. and is evaluated via the change of variable

$$t = \frac{1}{\sigma_{B|\Omega}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{B|\Omega}}{\sigma_{B|\Omega}}\right)^2\right] \quad ,$$

which yields

$$\int_{x=-\infty}^{x=b} f(x)\,dx =$$

$$\int_{t=0}^{t=\frac{1}{\sigma_{B|\Omega}\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{b-\mu_{B|\Omega}}{\sigma_{B|\Omega}}\right)^2\right]} f\left(\mu_{B|\Omega} - \sigma_{B|\Omega}\sqrt{-2\ln(\sigma_{B|\Omega}\sqrt{2\pi}t)}\right) \cdot \frac{\sigma_{B|\Omega}}{t}\sqrt{-2\ln(\sigma_{B|\Omega}\sqrt{2\pi}t)}\,dt$$

for $b \leq \mu_{B|\Omega}$, and

$$\int_{x=a}^{x=\infty} f(x)\,dx =$$

$$\int_{t=0}^{t=\frac{1}{\sigma_{B|\Omega}\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{a-\mu_{B|\Omega}}{\sigma_{B|\Omega}}\right)^2\right]} f\left(\mu_{B|\Omega} - \sigma_{B|\Omega}\sqrt{-2\ln(\sigma_{B|\Omega}\sqrt{2\pi}t)}\right) \cdot \frac{\sigma_{B|\Omega}}{t}\sqrt{-2\ln(\sigma_{B|\Omega}\sqrt{2\pi}t)}\,dt$$

for $a \geq \mu_{B|\Omega}$. The a_i's are chosen from $\{\mu_A - \sigma_A, \mu_A + \sigma_A, \mu_B - \sigma_B, \mu_B + \sigma_B, \mu_{B|\Omega} - \sigma_{B|\Omega}, \mu_{B|\Omega} + \sigma_{B|\Omega}\}$ to satisfy $-\infty < a_0 < \cdots < a_N < x$. The integral I is then evaluated using Gaussian quadrature [64] via Gauss-Hermite polynomials.

The integral

$$\frac{1}{K^{(\mathbf{b})}} \triangleq \int_{-\infty}^{\infty} g_\Omega(x)[F(x)]^{n_X^{(\mathbf{b})}}[G(x)]^{n_Y^{(\mathbf{b})}-1}\,dx$$

may be expressed as

$$\frac{1}{K^{(\mathbf{b})}} = I_1(-\infty, a_0) + \sum_{i=1}^{N} I_1(a_{i-1}, a_i) + I_1(a_N, \infty) \quad .$$

and each integral is evaluated using Romberg integration, with the a_i's chosen as above. The first and last integrals are both improper, and are evaluated using the above change of variable technique. Finally, the integral

$$\frac{1}{K^{(\mathbf{a})}} = \int_{-\infty}^{\infty} f_\Omega(x)[F(x)]^{n_X^{(\mathbf{a})}-1}[G(x)]^{n_Y^{(\mathbf{a})}}\,dx$$

may be expressed as

$$\frac{1}{K^{(\mathbf{a})}} = I_2(-\infty, a_0) + \sum_{i=1}^{N} I_2(a_{i-1}, a_i) + I_2(a_N, \infty) \quad ,$$

where

$$I_2(t_0, t_f) \triangleq \int_{t_0}^{t_f} f_\Omega(s)[F(s)]^{n_X^{(\mathbf{b})}-1}[G(s)]^{n_Y^{(\mathbf{b})}}\,ds \quad .$$

and each integral is evaluated using Romberg integration, with the a_i's chosen from $\{\mu_A - \sigma_A, \mu_A + \sigma_A, \mu_B - \sigma_B, \mu_B + \sigma_B, \mu_{A|\Omega} - \sigma_{A|\Omega}, \mu_{A|\Omega} + \sigma_{A|\Omega}\}$ to satisfy $-\infty < a_0 < \cdots < a_N < x$. Again, The first and last integrals are both improper, and are evaluated via the change of variable

$$t = \frac{1}{\sigma_{A|\Omega}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x - \mu_{A|\Omega}}{\sigma_{A|\Omega}}\right)^2\right] \quad .$$

Bibliography

1. Aho, Alfred B., et al. *Data Structures and Algorithms.* Reading, Massachusetts: Addison-Wesley, 1983.

2. Allen, Arnold O. *Probability, Statistics, and Queueing Theory: With Computer Science Applications.* Computer Science and Scientific Computing. San Diego, California: Academic Press, Inc., 1990.

3. Apostol, Tom M. *Mathematical Analysis* (Second Edition). Reading, Massachusetts: Addison-Wesley, 1974.

4. Arnold, Barry C., et al. *A First Course in Order Statistics.* New York: Wiley. 1992.

5. Bäck, Thomas. "Generalized Convergence Models for Tournament- and (μ, λ)-Selection." In Eshelman [19], 2–8.

6. Bäck, Thomas. *Evolutionary Algorithms in Theory and Practice.* New York: Oxford University Press, 1996.

7. Bäck, Thomas and Hans-Paul Schwefel. "An Overview of Evolutionary Algorithms for Parameter Optimization." *Evolutionary Computation, 1*(1):1–23 (1993).

8. Belew, Richard K. and Lashon B. Booker, editors. *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Mateo, California: Morgan-Kaufmann Publishers, July 1991.

9. Belew, Richard K. and Michael D. Vose, editors. *Foundations of Genetic Algorithms 4.* San Mateo, CA: Morgan Kaufmann, 1997.

10. Blanc-Lapierre, A. and R. Fortet. *Theory of Random Functions.* New York: Gordon and Breach, 1965.

11. Blickle, Tobias and Lothar Thiele. "A Mathematical Analysis of Tournament Selection." In Eshelman [19], 9–16.

12. Davidor, Yuval. "Epistasis Variance: A Viewpoint on GA-Hardness." In Rawlins [65], 23–35.

13. Davidor, Yuval, et al., editors. *Parallel Problem Solving from Nature — PPSN III*, number 866 in Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1994.

14. Davis, Lawrence, editor. *Handbook of Genetic Algorithms.* London: Pitman, 1987.

15. de la Maza, Michael and Bruce Tidor. "An Analysis of Selection Procedures with Particular Attention Paid to Proportional and Boltzmann Selection." In Forrest [25], 124–131.

16. Deb, Kalyonmoy. *Binary and Floating Point Optimization Using Messy Genetic Algorithms.* PhD dissertation, Department of Engineering Mechanics, The University of Alabama, Tuscaloosa, AL 35487-2908, April 1991.

17. DeJong, Kenneth A. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems..* PhD dissertation, The University of Michigan, Ann Arbor MI, 1975.

18. Ebeling, W. and H.-P. Schwefel, editors. *Parallel Problem Solving from Nature — PPSN IV*, 1996. In Preparation.

19. Eshelman, Larry J., editor. *Proceedings of the Sixth International Conference on Genetic Algorithms*, San Mateo CA: Morgan Kaufmann Publishers, Inc., 1995.

20. Fogel, D. B. and W. Atmar, editors. *Proceedings of the First Annual Conference on Evolutionary Programming.* La Jolla, CA: Evolutionary Programming Society, 1992.

21. Fogel, D. B. and W. Atmar, editors. *Proceedings of the Second Annual Conference on Evolutionary Programming.* La Jolla, CA: Evolutionary Programming Society, 1993.

22. Fogel, David, editor. *Proceedings of the Second IEEE Conference on Evolutionary Computation*, Piscataway NJ: IEEE Service Center, 1995.

23. Fogel, L. J., et al., editors. *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, Cambridge, MA: MIT Press, 1996.

24. Fogel, L. J., et al. *Artificial Intelligence Through Simulated Evolution*. New York: Wiley Publishing, 1966.

25. Forrest, Stephanie, editor. *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo CA: Morgan Kaufmann Publishers, Inc., July 1993.

26. Garey, M.R. and D.S. Johnson. *Computers and Intractibility: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

27. Gates, Jr., George H. *Predicting Protein Structure Using Parallel Genetic Algorithms*. MS thesis, AFIT/GCS/ENG/94D-03, Graduate School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH 45433, December 1994.

28. Gates, Jr., George H., et al. "Parallel Simple and Fast Messy GAs for Protein Structure Prediction." *Proceedings of the Intel Supercomputer Users' Group 1995 Annual North America Users Conference*. Beaverton, Oregon: Intel Supercomputer Systems Division, 1995.

29. Goldberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading MA: Addison–Wesley Publishing Company, 1989.

30. Goldberg, David E. Personal communication, Sept 1995.

31. Goldberg, David E., et al. "Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale," *Complex Systems*, *4*:415–444 (1990).

32. Goldberg, David E., et al. "Don't Worry, Be Messy." In Belew and Booker [8].

33. Goldberg, David E. and Kalyanmoy Deb. "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms." In Rawlins [65], 69–93.

34. Goldberg, David E., et al. "Genetic Algorithms, Noise, and the Sizing of Populations," *Complex Systems*, *6*:333–362 (1992).

35. Goldberg, David E., et al. "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms." In Forrest [25], 56–64.

36. Goldberg, David E., et al. "Messy Genetic Algorithms: Motivation, Analysis, and First Results," *Complex Systems*, *3*:493–530 (1989).

37. Grefenstette, John J., editor. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers, July 1985.

38. Grefenstette, John J., editor. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers, July 1987.

39. Grefenstette, John J. "Deception Considered Harmful." In Whitley [75].

40. Grefenstette, John J. and James E. Baker. "How Genetic Algorithms Work: A Critical Look at Implicit Parallelism." In Schaffer [67], 20–27.

41. Hogg, Robert V. and Allen T. Craig. *Introduction to Mathematical Statistics* (Fourth Edition). New York: Macmillan, 1978.

42. Holland, John H. "Outline for a Logical Theory of Adaptive Systems," *Journal of the Association for Computing Machinery*, *9*(3):297–314 (1962).

43. Holland, John H. *Adaptation in Natural and Artificial Systems*. Ann Arbor MI: The University of Michigan Press, 1975.

44. Holland, John H. *Adaptation in Natural and Artificial Systems* (First MIT Press Edition). Cambridge, MA: MIT Press, 1992.

45. Kagupta, Hillol. "The Gene Expression Messy Genetic Algorithm." In Thomas Bäck and Zbigniew Michalewicz and Hiroaki Kitano [72], 631–636.

46. Kargupta, Hillol. "Personal communication." Tournament selection parameters in the fast messy GA, December 1993.

47. Kargupta, Hillol. *SEARCH, Polynomial Complexity, and the Fast Messy Genetic Algorithm.* PhD dissertation, University of Illinois, 1995. Also available as IllliGAL Report No. 95008.

48. Lamont, Gary B., editor. *Compendium of Parallel Programs for the Intel iPSC Computers.* Wright-Patterson AFB OH 45433: Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, 1993.

49. Liepins, Gunar E. and Michael D. Vose. "Deceptiveness and Genetic Algorithm Dynamics." In Rawlins [65], 36–50.

50. Luenberger, David G. *Optimization by Vector Space Methods.* New York: Wiley. 1969.

51. Männer, Reinhard and Bernard Manderick, editors. *Parallel Problem Solving from Nature, 2*, Amsterdam: North-Holland, September 1992.

52. McDonnell, J. R., et al., editors. *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, Cambridge, MA: MIT Press, 1995.

53. Mendenhall, William, et al. *Mathematical Statistics with Applications* (Fourth Edition). Boston: PWS-KENT Publishing Company. 1990.

54. Merkle, Laurence D., et al. "Application of the Parallel Fast Messy Genetic Algorithm to the Protein Folding Problem." *Proceedings of the Intel Supercomputer Users' Group 1994 Annual North America Users Conference.* 189–195. Beaverton Oregon: Intel Supercomputer Systems Division. 1994.

55. Merkle, Laurence D., et al. "Hybrid Genetic Algorithms for Minimization of a Polypeptide Specific Energy Model." In Thomas Bäck and Zbigniew Michalewicz and Hiroaki Kitano [72], 396–400.

56. Merkle, Laurence D., et al. "Hybrid Genetic Algorithms for Polypeptide Energy Minimization." *Applied Computing 1996: Proceedings of the 1996 Symposium on Applied Computing.* 396–400. New York: The Association for Computing Machinery. 1996.

57. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs* (Third Edition). Springer, 1996.

58. Michalewicz, Zbigniew. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer. 1992.

59. Michalewicz, Zbigniew, editor. *Proceedings of the First IEEE Conference on Evolutionary Computation.* Piscataway NJ: IEEE Service Center. June 1994.

60. Mitchell, Melanie. *An Introduction to Genetic Algorithms.* Cambridge, MA: The MIT Press, 1996.

61. Naylor, Arch W. and George R. Sell. *Linear Operator Theory in Engineering and Science. 40.* Applied Mathematical Sciences. New York: Springer-Verlag, 1982.

62. Pearl, Judea. *Heuristics.* Reading MA: Addison-Wesley Publishing Company. 1989.

63. Pierre, Donald A. *Optimization Theory with Applications.* New York: Dover. 1986.

64. Press, William H., et al. *Numerical Recipes in C: The Art of Scientific Computing* (Second Edition). New York: Cambridge University Press, 1992.

65. Rawlins, Gregory J. E., editor. *Foundations of Genetic Algorithms.* San Mateo, CA: Morgan Kaufmann, 1991.

66. Rechenberg. Ingo. *Evolutionsstratrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.

67. Schaffer. J. David, editor. *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, California: Morgan-Kaufmann Publishers, Inc., June 1989.

68. Schwefel, H.-P. and R. Männer, editors. *Parallel Problem Solving from Nature, 496*. Lecture Notes in Computer Science, Berlin: Springer-Verlag, October 1990.

69. Schwefel, Hans-Paul. *Numerische Optimierung von Computer-Modellan mittels der Evolutionsstrategie*. New York: Wiley, 1981.

70. Schwefel, Hans-Paul. *Evolution and Optimum Seeking*. New York: Wiley, 1995.

71. Sebald. A. V. and L. J. Fogel, editors. *Proceedings of the Third Annual Conference on Evolutionary Programming*, River Edge, NJ: World Scientific, 1994.

72. Thomas Bäck and Zbigniew Michalewicz and Hiroaki Kitano, editor. *Proceedings of the Third IEEE Conference on Evolutionary Computation*, Piscataway NJ: IEEE Service Center, 1996.

73. Vásquez, Maximiliano, et al. "Conformational Energy Calculations on Polypeptides and Proteins," *Chemical Reviews*, *94*:2183–2239 (1994).

74. Vose, Michael D. and Alden H. Wright. "Simple Genetic Algorithms with Linear Fitness," *Evolutionary Computation*, *2*(4):347–368 (1994).

75. Whitley, L. Darrell, editor. *Foundations of Genetic Algorithms 2*. San Mateo, CA: Morgan Kaufmann, 1993.

76. Whitley. L. Darrell and Michael D. Vose, editors. *Foundations of Genetic Algorithms 3*. San Mateo, CA: Morgan Kaufmann, 1995.

Vita

Laurence D. Merkle spent his youth as an "Air Force brat," mostly in Albuquerque, New Mexico. He earned his bachelor's degree in Computers and Systems Engineering from the Rensselaer Polytechnic Institute in 1987. At the same time, he earned his commission through Air Force ROTC. Upon entering active duty, he was assigned to the Air Force Logistics Command's Artificial Intelligence Program Management Office, at Wright-Patterson AFB, Ohio. While there, he taught introductory AI classes and trained AFLC personnel in the development of small rule-based expert systems. In 1991 Captain Merkle married Margaret Tossey, a Dayton native. Shortly thereafter, he was assigned to the Air Force Institute of Technology to earn his Masters degree in Computer Engineering. His thesis research investigated mappings of messy genetic algorithms to coarse-grained parallel computer architectures. Upon completion of his Masters degree, he remained at AFIT to earn his doctorate. In that role he published a number of papers, mainly related to the application of parallel genetic algorithms to polypeptide structure prediction. Upon defending this dissertation, he returned to Albuquerque as a group leader in the Air Force Materiel Command's Center for Plasma Theory and Computation, at Kirtland AFB.

Permanent address: `l.merkle@ieee.org`
5439 Thomas Dr NE
Albuquerque, New Mexico 87111

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1996

3. REPORT TYPE AND DATES COVERED
PhD Dissertation

4. TITLE AND SUBTITLE
Analysis of Linkage-Friendly Genetic Algorithms

5. FUNDING NUMBERS

6. AUTHOR(S)
Laurence D. Merkle, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/DS/ENG/96-11

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Wright Laboratory (AFMC)
Materials Directorate
Wright-Patterson AFB, OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Evolutionary algorithms (EAs) are stochastic population-based algorithms inspired by natural selection, mutation, and recombination, often employed as optimum seeking technques. A formal framework is proposed. Evolutionary operators are viewed as mappings from parameters to random functions. Formal definitions capture the distinguishing characteristics of operator types. EAs using strictly invariant selection and order-invariant representations are linkage-friendly genetic algorithms (lfGAs).

Fast messy genetic algorithms (fmGAs) are lfGAs which use binary tournament selection (BTS) with thresholding, periodic filtering of random genes, and generalized single-point crossover. EAs using probabilistic variants of thresholding and filtering are generalized fmGAs (gfmGAs).

A dynamical systems model is developed and predicts expected effectiveness. BTS with probabilistic thresholding is modeled as a hierarchical Markov chain. Transitions at the lowest level involve decisions between classes. The probability of correct decision making is related to appropriate maximal order statistics. Filtering models are extended to include probabilistic lengths.

Sensitivity of lfGA effectiveness to parameters limits practical applications. The parameter selection problem is formally posed as a constrained optimization problem. The cost functional is related to expected effectiveness. Kuhn-Tucker conditions are derived. Parameter selection techniques are proposed for fmGAs and gfmGAs.

14. SUBJECT TERMS
Evolutionary Computation, Genetic Algorithms, Optimization

15. NUMBER OF PAGES
159 + 12

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL